Maurizio Galluzzo
il Manuale dell'ATARI 520 ST e 1040 ST
HOME E PERSONAL COMPUTER
GRUPPO EDITORIALE JACKSON
ATARI
520ST

Maurizio Galluzzo

il Manuale dell'ATARI 520 ST e 1040 ST

GRUPPO
EDITORIALE
JACKSON
Via Rosellini, 12
20124 Milano

Prefazione

I computer Atari 520 ST e 1040 ST costituiscono una rivoluzione nel campo dei personal computer per la loro versatilità e potenza. Il dialogo con la macchina avviene per mezzo di immagini (icone) e del MOUSE la cui posizione sul piano di lavoro è strettamente legata a quella sullo schermo. Si tratta di un metodo di colloquio con la macchina particolarmente apprezzato da programmatori professionisti e utilizzatori (anche alle prime armi). Il computer ha inoltre capacità grafiche, cromatiche e sonore superiori a quelle di qualsiasi personal attualmente esistente nel mercato. Queste caratteristiche unitamente ad un prezzo eccezionalmente competitivo e alla quantità di software disponibile lo pongono in primo piano come computer ad uso professionale ed aziendale. Nonostante la forte concorrenza in questo campo e alla relativa gioventù del sistema, credo non potrà avere rivali anche considerando un utilizzo esasperatamente professionale. Veniamo ora al libro destinato a vari gradi di utenza , dai professionisti preparati, a coloro che si trovano per la prima volta davanti ad un computer. A questi ultimi consiglio però anche la lettura di un manuale di BASIC (di questo genere, l' Editore presenta un buon numero di titoli destinati a diversi gradi di preparazione). Dopo la descrizione sommaria delle caratteristiche del computer e all'utilizzo dell' interfaccia utente-macchina per mezzo di icone, si passa ad una rapida spiegazione sui linguaggi di programmazione e ad un piccolo corso di BASIC. Piatto forte del libro sono i programmi originali scritti in BASIC che meglio di molte parole possono spiegare il funzionamento dei comandi del linguaggio. Consiglio quindi una accurata lettura dei listati ampiamente commentati con elenco variabili e singole istruzioni.

L'uso della macchina è facile ed intuitivo ma per un utilizzo più sicuro vi invito alla lettura del testo cercando di realizzare in proprio esempi sugli argomenti trattati. Voglio ringraziare il Sig. Veronesi Direttore Commerciale della ATARI ITALIA SPA di Milano e il Dr. Capone della Inter Service di Padova per la collaborazione offerta. Su queste righe auguro a tutti buon lavoro.

Maurizio Galluzzo

Indice

IL COMPUTER ATARI 520 ST E 1040 ST

La realizzazione di computer a basso costo ha rivoluzionato notevolmente il nostro modo di operare non solo in ufficio e nei posti di lavoro ma anche in casa, nella nostra corrispondenza, nei nostri rapporti con gli altri. Questi argomenti non sono certamente nuovi ma è il caso di sottolinearli in quanto enfatizzati ma mai capiti dai mass media. Questo libro ha l'intento di facilitarvi l'uso del computer Atari 520 ST e 1040 ST: macchine rivoluzionarie nei contenuti e nelle caratteristiche di interfaccia uomo-computer. Si tratta inoltre di strumenti completi dal punto di vista hardware e software.

Iniziamo questa nostra chiacchierata introduttiva con la spiegazione di questi termini entrati ormai nel linguaggio quotidiano.

Con il termine HARDWARE si intende la parte fisica della macchina: il computer e le periferiche ad esso collegate (stampante, unità a dischi, plotter ...). Il termine significa testualmente "ferraglia".

SOFTWARE sta invece a significare i programmi, sono le istruzioni che permettono al computer di eseguire tutti i compiti richiesti. Il software può essere RESIDENTE cioè all'interno della macchina oppure realizzato dall'utente per le sue attività.

Un computer è costituito da una serie di parti che potremmo così distinguere:
- UNITÀ CENTRALE (CPU) costituita dal PROCESSORE e dalla MEMORIA PRINCIPALE
- UNITÀ DI INGRESSO (INPUT)
- UNITÀ DI USCITA (OUTPUT)
- MEMORIA DI MASSA

Nel nostro caso il processore è un MC68000 a 32 bit interni e 16 bit esterni. La memoria principale e ROM (memoria di sola lettura) e RAM (memoria di lettura e scrittura). Le unità di ingresso dei dati sono: la tastiera, il mouse, il joystick eccetera.

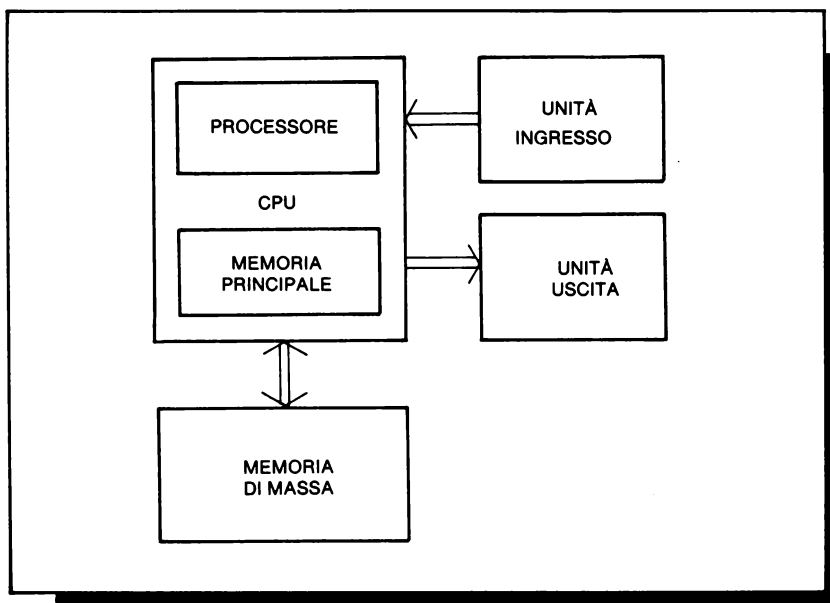



Fig. 1.

Le unità di uscita sono costituite dal monitor (b/n e colore) , dalle stampanti eccetera. La memoria secondaria è costituita dai floppy disk e hard disk.
Abbiamo parlato di "bit" si tratta della contrazione delle parole "binary digit" cioè cifra binaria. Possono assumere solo due valori 0 o 1 (falso o vero, spento o acceso eccetera). Il sistema di numerazione binario è l'unico che viene riconosciuto dal processore e quindi tutte le nostre operazioni verranno trasformate in questa notazione. Naturalmente scrivere un programma in questo modo sarebbe impossibile. Sono stati realizzati quindi dei linguaggi che si avvicinano maggiormente a quello naturale e sono quindi più vicini al modo di pensare dell'uomo. I linguaggi realizzati sono numerosi e presentano vantaggi e svantaggi rispetto all'utilizzo che se ne intende fare. Di questi argomenti parleremo comunque in un'altro capitolo del libro.

La memoria RAM è volatile ovvero si cancella non appena la macchina viene spenta. Occorre quindi salvare i nostri programmi e dati in una maniera più duratura. Questo avviene per mezzo della MEMORIA DI MASSA il cui accesso è più lento ma duraturo.

Noi useremo l'unità a dischi che è attualmente la più versatile ed economica.

I dischi si qualificano in base alle loro dimensioni. Quelli usati dagli ATARI 520 ST e 1040 ST sono da 3 pollici e mezzo e hanno una custodia rigida.

Veniamo ora ad esaminare i componenti del nostro sistema .

La tastiera

È composta da 94 tasti che possono essere divisi in tasti funzione, tasti del pad numerico e tastiera vera e propria. I tasti funzione sono numerati (da <F1> a <F10>) e assolvono, come dice il loro stesso nome, ad una particolare attività a seconda del programma inserito. Il tastierino numerico alla destra ha le stesse funzioni dei tasti con i numeri ma è di più facile utilizzo. I tasti con le frecce servono a muovere il cursore nello schermo nella fase di <EDIT> dei programmi cioè nella fase di stesura e correzione degli stessi.

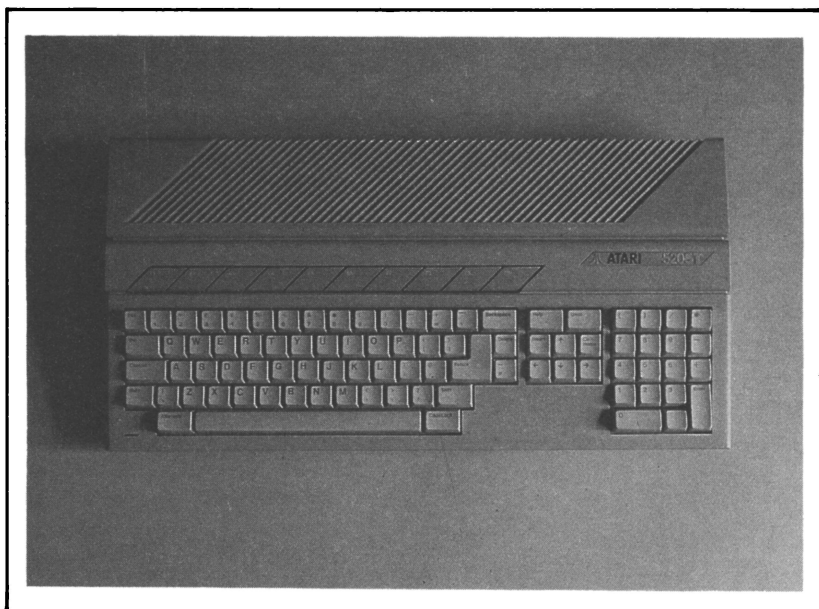



Fig. 2.

Nella parte posteriore del computer troviamo una serie di interruttori e connettori: esaminiamoli da sinistra a destra.

RESET — Serve a resettare il computer. Cancella tutto il contenuto della memoria e deve essere usato solo quando il computer entra in crash ovvero è impossibile continuare qualsiasi operazione e la tastiera risulta disabilitata. Ha lo stesso effetto di togliere per qualche istante l'alimentazione.

ON/OFF — Interuttore di accensione del computer.
L'alimentazione dello stesso è segnalata dall'accensione di un led rosso nella parte in basso a sinistra della tastiera (accanto al tasto ALTERNATE).

POWER	A questo connettore va collegato il cordone di alimentazione del computer. Il corretto inserimento è guidato da due riferimenti.
MIDI OUT/IN	Il termine MIDI significa Musical Instrument Digital Interface. Questa porta consente il collegamento con strumenti musicali professionali. L'interfaccia permette quindi il controllo di apparecchiature elettroniche particolarmente sofisticate per la creazione di suoni e di musica.
MONITOR	Permette il collegamento di un monitor ad alta risoluzione a colori o in bianco e nero.
PRINTER	Questo connettore consente il collegamento con una stampante con interfaccia parallela. Le stampanti dedicate sono anche in grado di eseguire l'hard copy dello schermo ovvero la copia punto per punto dello schermo del monitor.
MODEM	A questo connettore può essere attaccato un modem, uno strumento cioè in grado di collegare, per mezzo del telefono, il tuo computer con un'altro, posto anche a parecchi chilometri di distanza. Invece del modem può essere connessa una stampante parallela o qualsiasi altro strumento che utilizzi l'interfaccia RS232C. Questa interfaccia è sicuramente tra le più diffuse nelle connessioni tra computer. L'uso del modem si sta rilevando in questi ultimi tempi in tutta la sua importanza specie nel collegamento con banche di dati, servizi informativi, posta elettronica e altro. Questa branchia dell'informatica, che si chiama TELEMATICA, avrà nei prossimi anni sempre maggiore sviluppo e costituirà la vera rivoluzione del lavoro post-industriale. Si potrà lavorare tranquillamente in casa e al momento voluto, senza orari e quindi inviare in ufficio i testi scritti o le pratiche evase.
FLOPPY DISK	Abbiamo già detto che i dischi usati sono da 3 pollici e mezzo. Ogni singola faccia del dischetto è in grado di memorizzare fino a 360 mila Byte. Un Byte è una unità di misura della memoria ed è composto da un certo numero di bit. Al computer possono essere connessi fino ad un massimo di due MicroFloppy Disk Drive.
HARD DISK	Quando è necessario archiviare un gran numero di informazioni ed averle a disposizione in tempi brevi si ricorre all'uso del disco rigido (Hard Disk) che è in grado di memorizzare dati ad alta velocità. La velocità di trasferimento della porta per HD è di circa 10 milioni di bit al secondo. La capacità di memoria di un disco rigido arriva ad alcune decine di milioni di caratteri.

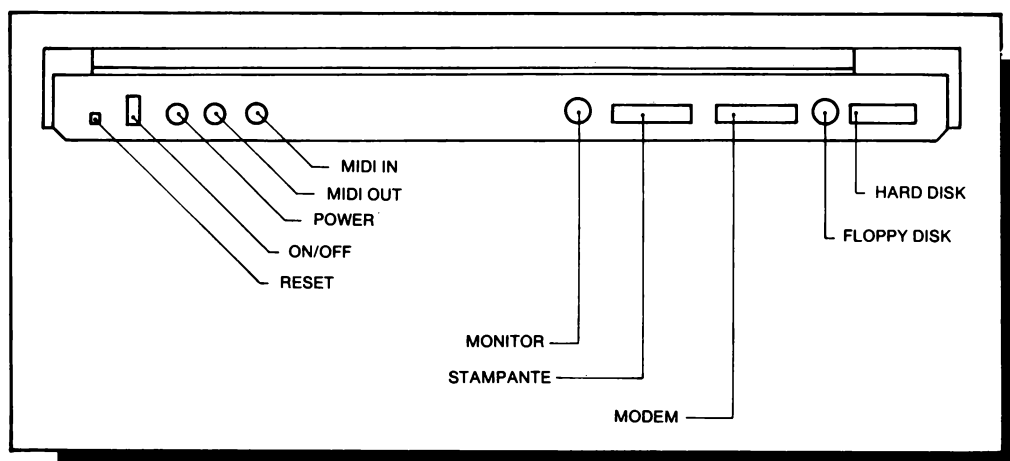



Fig. 3.

CARTRIDGE — Nella parte sinistra del computer troviamo una apertura nella quale è possibile inserire una cartuccia (CARTRIDGE). La cartuccia contiene programmi in ROM (memoria di sola lettura). Ogni cartuccia può contenere fino a circa 130 KByte di memoria. Il prefisso <K> sta ad indicare mille (per la precisione 1024).

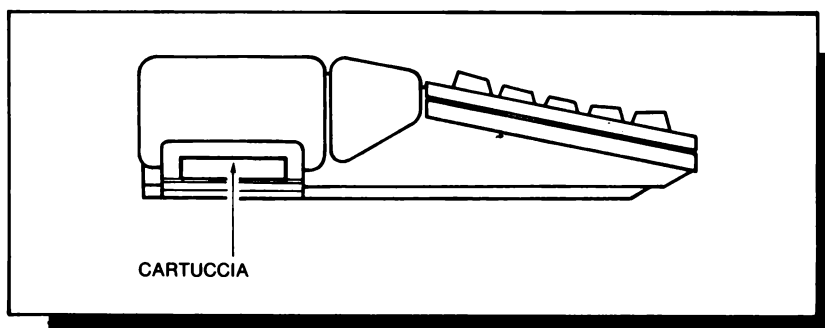



Fig. 4.

MOUSE — Nella parte sinistra della macchina sono presenti due porte, quella più a sinistra denominata <0> e quella più a destra <1>. Alla porta <0> deve essere connesso il mouse.

JOYSTICK — Indifferentemente alla porta <0> oppure <1> possono essere connessi dei joystick per il controllo esterno dei programmi (non solo di giochi).

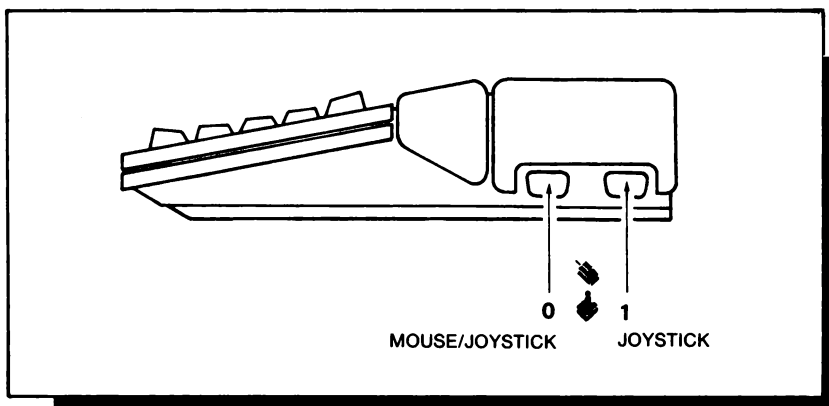



Fig. 5.

NOTA IMPORTANTE. Sopra alla tastiera sono presenti alcune ferritoie di raffreddamento. È importante durante l'uso della macchina che tali aperture non siano coperte da fogli, carte, libri eccetera, in quanto è necessario per un corretto funzionamento un raffreddamento costante dei circuiti interni. È altresì importante sottolineare che i connettori non dovranno essere forzati durante l'inserimento e non dovranno essere toccati con le mani o con arnesi metallici.

Il mouse

Il mouse (in inglese 'topo') è lo strumento per il controllo di tutte le funzioni del computer. Si tratta di un comando a distanza la cui posizione sul piano di scorrimento ha un riferimento sullo schermo del computer. Le immagini che appaiono sul monitor sono chiamate 'icone' ad esse viene fatto riferimento durante l'uso. L'idea è quella di indicare con una freccia la funzione che si intende utilizzare. In questo modo l'uso del computer è esteso anche a coloro che si trovano per la prima volta davanti alla macchina. Questo modo di utilizzo simbolico (non si può ancora parlare di programmazione) è molto facile ed intuitivo. Evita infatti di imparare decine di comandi (spesso in inglese) per compiere le funzioni più consuete. Questo modo di operare sarà sempre più frequente nelle macchine da destinare all'ufficio. Il mouse è costituito da una sfera gommata che viene fatta scorrere sul piano di lavoro e la cui rotazione (quindi posizione) viene rilevata da tre sensibili sensori. Tali sensori trasferiscono i valori registrati in coordinate cartesiane riferite

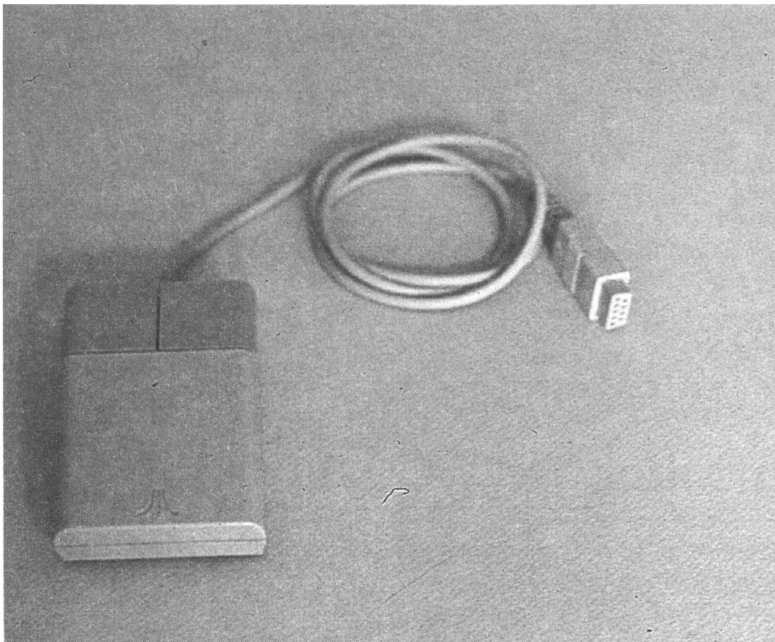



Fig. 6. Il mouse.

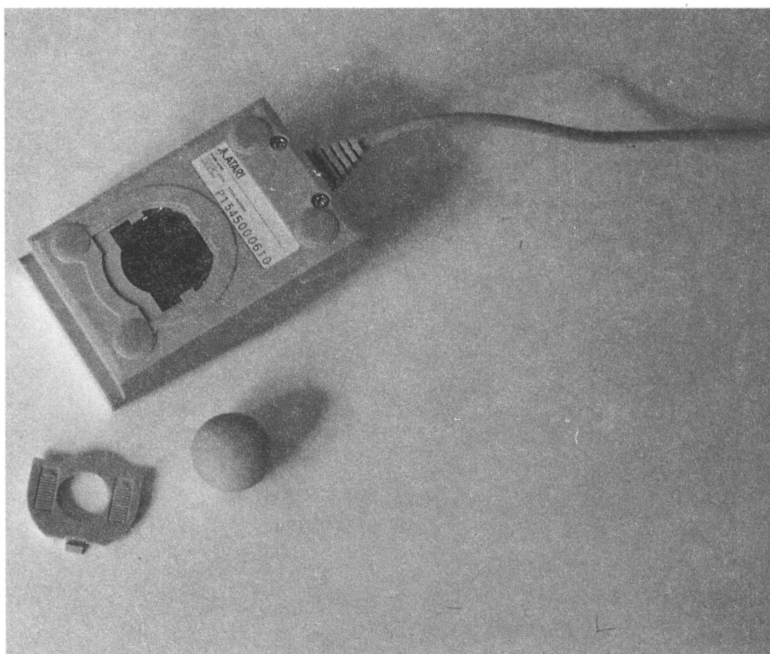



Fig. 7. Il mouse con la sfera posizionale in evidenza.

agli assi dello schermo. Capovolgendo il mouse potrete scorgere la sfera in questione. Maggiori informazioni sono presenti nel capitolo dedicato alla manutenzione (cap. 3). Sopra al mouse sono presenti due tasti. Voi dovrete usare esclusivamente quello di SINISTRA. L'altro viene attivato solo in alcuni programmi. Il tasto serve a confermare una scelta a 'prendere' una icona eccetera.

La posizione del mouse è normalmente sulla sinistra della tastiera e dovrà essere fatto scorrere su una superficie liscia e pulita.

Il monitor

Il monitor usato nel collegamento con il computer è ad alta risoluzione ovvero è in grado di visualizzare molte linee senza introdurre deformazioni o distorsioni dell'immagine.

Il monitor utilizzato può essere monocromatico (bianco e nero) oppure a colori. Al monitor è incorporato anche un altoparlante per la emissione dei suoni prodotti dalla macchina. Dopo aver collegato il

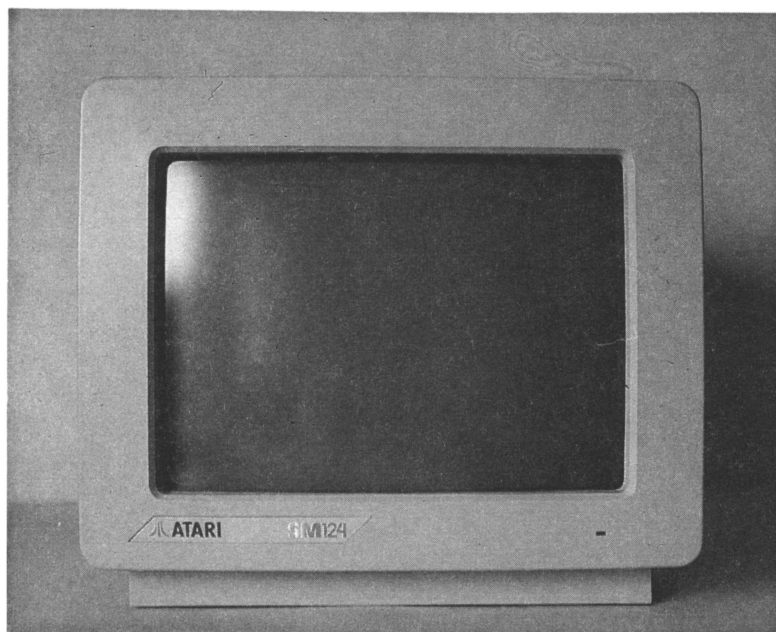



Fig. 8. Monitor monocromatico SM 124.

monitor all'alimentazione, lo si può accendere agendo sul comando più in basso che si trova sulla destra dello schermo. Lo stesso comando funziona anche da controllo del volume. Il comando centrale stabilisce il contrasto dell'immagine e quello più in alto la luminosità.

L'alimentazione al monitor è indicata con l'accensione di un led verde nella parte in basso a destra.

Riassumiamo i comandi del monitor dall'alto in basso:
* LUMINOSITÀ
* CONTRASTO
* ON/OFF E VOLUME

Disk Drive

Il sistema può controllare fino a due disk drive.

Il disk drive, come si è già avuto modo di dire, è in grado di memorizzare dati, programmi e documenti in dischi da 3 pollici e mezzo. I dichetti dovranno essere inseriti con la scritta verso l'alto e non dovranno essere posti sopra fonti di calore o campi magnetici molto forti. È bene anche

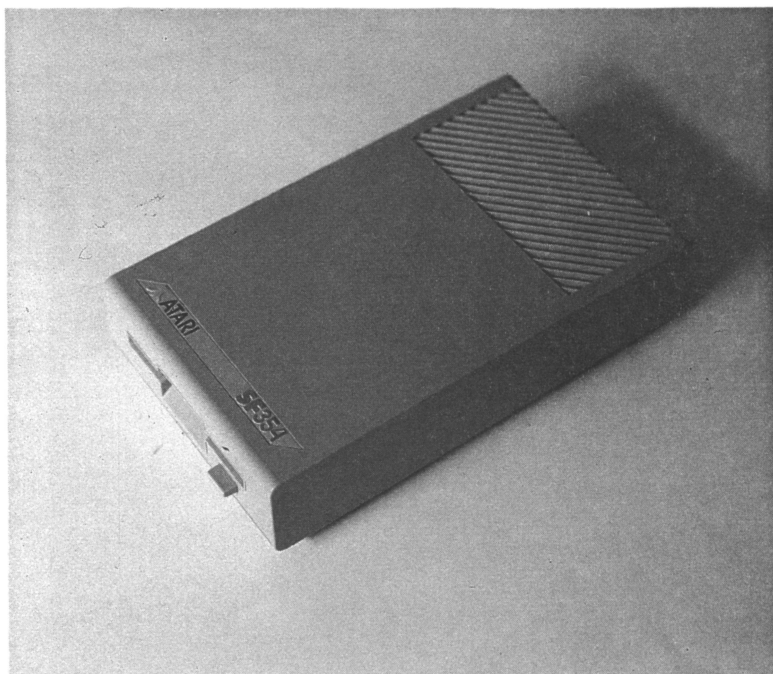



Fig. 9. Disk Drive SF 354.

riporli in appositi contenitori in modo da non essere esposti alla polvere o alla umidità che potrebbe danneggiarli facendovi perdere anche importanti dati. Per estrarre il disco si dovrà agire sull'apposito comando posto davanti al disk drive. Il disk drive dovrà essere collegato all'alimentatore e acceso tramite l'interruttore posto nella parte posteriore. Un led rosso indicherà quando il drive è in funzione (non quando è acceso) e quindi il disco non dovrà essere in alcun modo rimosso. La testina di lettura/scrittura potrebbe rovinarlo (e rovinarsi) irrimediabilmente. Il drive dovrà essere collocato preferibilmente sulla sinistra della tastiera per facilitare le operazioni di inserimento e disinserimento del disco stesso.

Un disco, dopo l'acquisto, non può essere utilizzato direttamente ma solo dopo essere stato FORMATTATO. Dopo questa operazione (che più avanti avremmo modo di spiegare in dettaglio) è bene sempre segnare il nome del disco stesso (o del suo contenuto) per evitare facili confusioni. Il dischetto dispone nella parte in basso a sinistra di una protezione che permette di impedire di cancellare per errore dei dati inseriti.

La protezione può essere attivata e rimossa agendo con una penna e spostando la linguetta scorrevole. Nella parte posterione del disk drive troviamo una serie di connettori.

Vediamoli in successione.

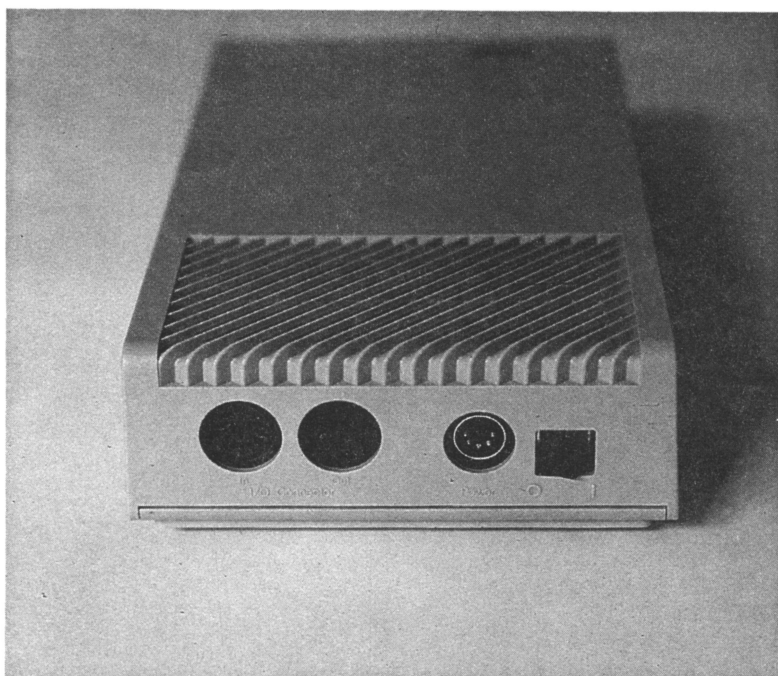



Fig. 10. Vista posteriore del drive SF 354.

- Il connettore di sinistra deve essere collegato al computer.
- Il secondo connettore serve solo se intendiamo collegare un altro disk drive. In questo caso si procederà come in fig. 11 .
- Il terzo connettore è quello che dovrà essere collegato con l'apposito alimentatore stabilizzato.
- All'estrema destra troviamo l'interruttore di alimentazione.

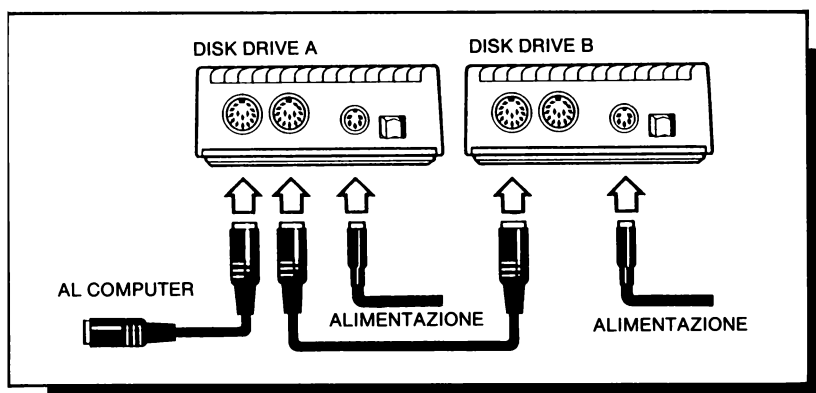



Fig. 11. Collegamento di due drives SF 354.

Installazione

Il personal computer è costituito nella sua configurazione più tipica, dalla tastiera, dal monitor e dalla unità a dischi. Una volta estratti i vari componenti dagli imballi è neccessario ricorrere ad una installazione e connessione degli stessi. È opportuno prestare particolare attenzione a questa fase per evitare malfunzionamenti e danni anche gravi. La prima cosa da fare è trovare un piano d'appoggio liscio e sufficiente per collocarvi tutti i componenti. L'altezza del piano deve essere adeguata a quella dell'utilizzatore. L' uso prolungato di un computer posto in posizione scomoda è particolarmente faticoso e può provocare danni alla vista e alla spina dorsale. Si dovrà porre la tastiera davanti al monitor, l'unità a dischi sulla sinistra e il mouse alla destra della tastiera stessa. La lunghezza del cordone del mouse è sufficiente per un utilizzo comodo anche per coloro che sono mancini. Oltre a questi componenti sono neccessari anche gli alimentatori (uno per il drive e l'altro per la tastiera) che andranno normalmente posti a terra anche per motivi estetici oltre che di disponibilità di spazio.

I collegamenti che dovranno essere scrupolosamente effettuati sono riportati nella figura 12.

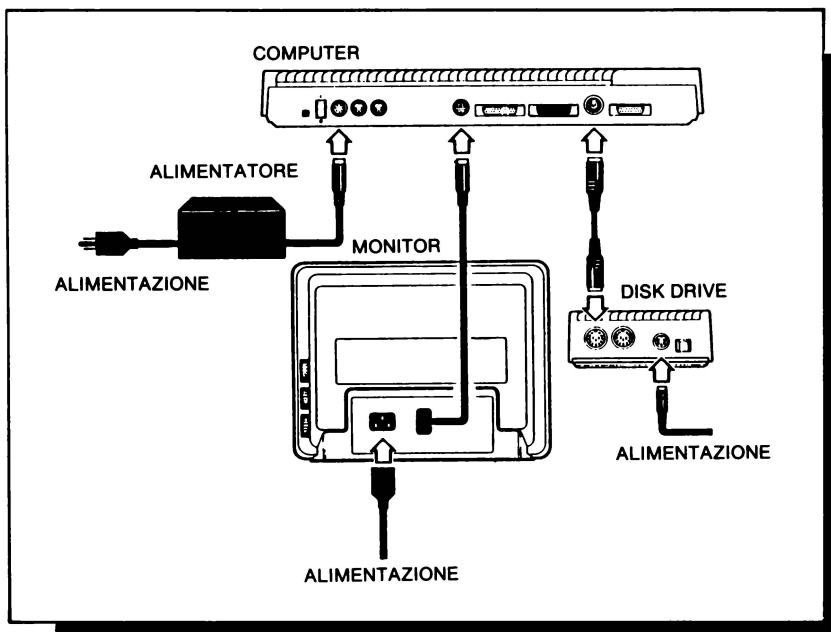



Fig. 12. Connessioni tra le varie parti.

Il mouse dovrà essere collegato alla porta più a sinistra delle due riportate accanto del pad numerico (a destra del computer).

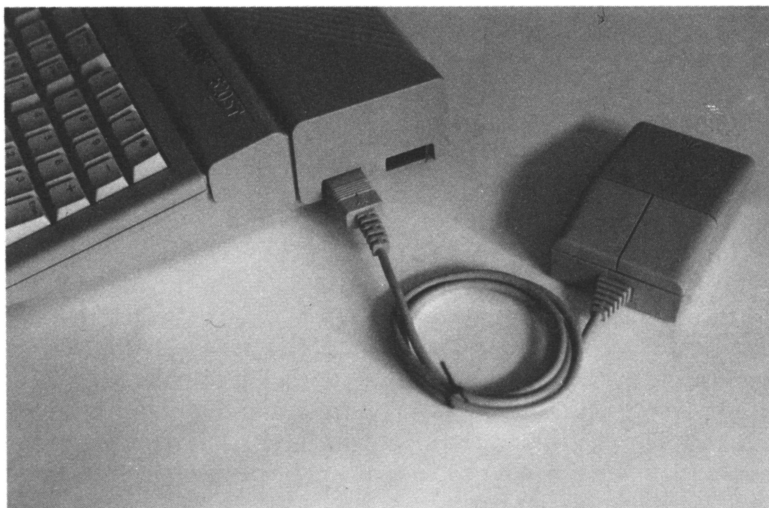



Fig. 13. Collegamento del mouse con il computer.

2

UTILIZZO DEL COMPUTER

Dopo l'installazione del computer è possibile iniziare ad utilizzarlo in maniera completa e con tutte le sue potenzialità. Il computer deve essere acceso unitamente all'unita a dischi e al monitor. Si raccomanda ancora una volta di controllare con cura la bontà e la correttezza delle connessioni. Sullo schermo del monitor apparirà al centro una piccola ape e dopo qualche istante un disegno con indicato di inserire il disco di sistema nel drive e quindi muovere la freccia sopra al rettangolo con l'indicazione 'OK'. Il tasto da usare è quello a sinistra del mouse.

Qualora l'immagine non dovesse apparire, controllare che i comandi del monitor della luminosità e del contrasto siano regolati in maniera opportuna.

Il disco da inserire (se non disponete della versione su ROM) è indicato dalla scritta 'TOS'. È importante non estrarre il disco quando la luce rossa del drive è accesa. Se avrete seguito correttamente le indicazioni fin qui descritte il computer inizierà a caricare il sistema operativo dal disco e alla fine sullo schermo appariranno una serie di 'oggetti' che vengono chiamati icone (Fig. 14).

L'ambiente di lavoro in cui ci si trova ad operare è denominato GEM (Graphics Environment Manager) ed è attualmente il più sofisticato e allo stesso tempo semplice modo di colloquio tra macchina e utente nel campo dei personal computer. Questo modo di operare presenta un enorme vantaggio rispetto ai sistemi tradizionali. Non è infatti neccessario imparare alcun comando per eseguire le funzioni più consuete e non esiste nemmeno il rischio di cancellare per errore ore di lavoro.

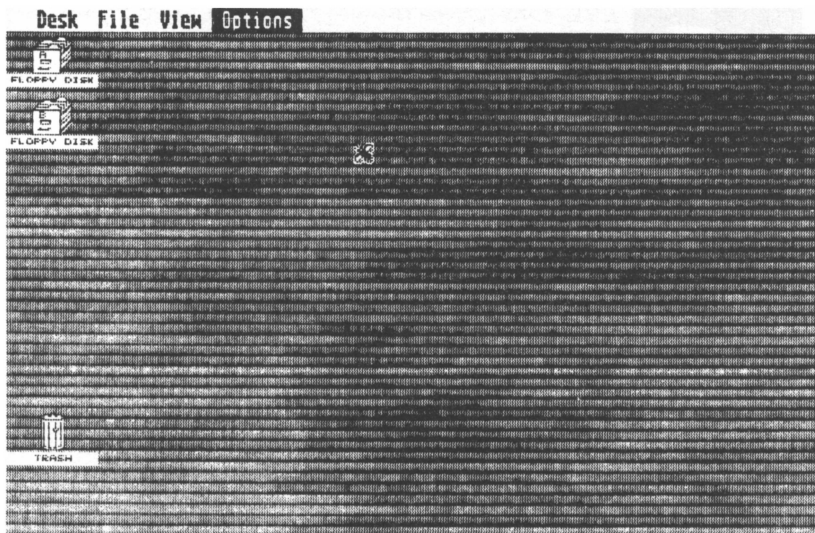



Fig. 14. Schermo dopo il caricamento del TOS.

Gem (Graphics Environment Manager)

Il GEM facilita e accellera tutte le possibilità di utilizzo della macchina. Lo strumento principe nel suo utilizzo è il mouse. Il mouse può muovere la freccia che appare sul video in ogni posizione dello schermo e nel suo uso si deve fare riferimento al punto indicato dall'estremità della freccia.

Nella parte alta dello schermo appare la barra dei menù nel quale sono indicati i titoli delle opzioni disponibili. Se si passa sopra a questa area con il mouse appariranno dei menù a discesa che potranno essere utilizzati selezionandoli con la freccia (cambieranno in REVERSE) e confermati con la pressione del tasto. Per chiuderli (senza alcuna scelta) e sufficiente allontanarsi dai menù stessi e premere il tasto del mouse. Le icone a sinistra indicano la presenza di due disk drive (se ne possedete solo uno potrete fare indifferentemente riferimento ad A o B). Nella parte bassa è indicato un contenitore dei rifiuti che serve per cancellare un documento, un programma o qualsiasi altro tipo di file creato. Le icone possono essere spostate nella loro posizione ponendo sopra il cursore e tenendo schiacciato il pulsante a sinistra. L'icona apparirà in REVERSE (con i colori scambiati) e contornata da un tratteggio. A questo punto (senza mai rilasciare il tasto) può essere trascinata in qualunque parte dello schermo (per il momento attenzione a non portarla sopra un' altra

icona). Per 'disattivarla' sarà sufficiente spostarsi da essa e premere il tasto. Il GEM permette lo spostamento di una serie di icone contemporaneamente. In questo caso è sufficiente portarsi a sinistra di esse e più in alto, tenere prenuto il tasto spostandosi in basso sinistra. Apparirà sullo schermo un rettangolo che circonderà le icone interessate. Rilasciate il tasto (le icone selezionate si coloreranno). A questo punto è sufficiente agganciarne una sola e trasportarla nel luogo voluto. Anche se a prima vista l'operazione può apparire difficile con qualche minuto di pratica risulterà molto istintiva. Lo spostamanto delle icone serve non solo per la copia dei dischi ma anche per evitare che siano sovrapposte alle finestre che verranno successivamente aperte. Oltre al mouse il cursore può essere mosso anche con la tastiera (vedi Cap. 2.3). Durante le prove che eseguirete sulle icone è consigliabile estrarre il disco dal drive per evitare spiacevoli e involontarie cancellazioni dei programmi contenuti.

Uso del mouse

Il piano di scorrimento del mouse deve essere liscio, asciutto e pulito. L'umidità è particolarmente temibile in quanto i sensori di posizione potranno essere danneggiati dalla presenza di liquidi nei loro meccanismi. In alcuni programmi il cursore può assumere forme diverse da quelle consuete della freccia oppure quella dell'ape. la freccia appare quando il cursore è attivo e può quindi compiere selezioni, movimenti eccetera. L'ape appare quando il computer disabilita il suo uso (ad esempio durante il caricamento di un disco oppure qundo si esegue una stampa). In gergo queste operazioni si chiamano di INPUT / OUTPUT (facendo riferimento alla direzione che assumono i dati). La rapidità di risposta del tasto del mouse può essere controllato per mezzo di un comando che si trova nel menù 'Desk'.

Uso della tastiera come mouse

Oltre che con il mouse (che è lo strumento naturale) il cursore può essere mosso tramite l'uso opportuno della tastiera. Vediamo di seguito la tabella riassuntiva (i tasti cursore sono quelli indicati con le frecce). Il simbolo '+' indica che i tasti devono essere prenuti contemporaneamente.

ALTERNATE + tasti cursore.................................... per muovere di 8 pixel
ALTERNATE + SHIFT + tasti cursore................ per muovere di un pixel
ALTERNATE + INSERT... sostituisce il tasto sinistro
ALTERNATE + CLR/HOME.................................. sostituisce il tasto destro

Naturalmente questi tasti si possono combinare. Ad esempio la funzione di scelta di più icone contemporaneamente può essere fatta premendo nello stesso momento ALTERNATE + INSERT + tasti cursore.

Le finestre

Una delle caratteristiche più particolari, oltre all'uso del mouse, è l'idea della finestra. Con questo termine si indica una porzione di schermo che viene gestita in maniera autonoma dalle altre. Un esempio può essere dato dall'avere una finestra contenente il listato di un programma e nell'altra l'esito dell'elaborazione. Le facilitazioni introdotte) specie in fase di debug, messa a punto dei programmi) sono immediate.

Vediamo ora come può essere trattata una finestra. Dopo aver caricato il TOS portate il cursore sull'icona DRIVE A. Questa operazione deve essere fatta senza togliere il disco di sistema dal drive stesso. Premete il tasto sull'icona stessa) andrà in reverse) e quindi scegliete il menù discendente 'File' e premete nuovamente il mouse su 'OPEN'. Noterete che alcune funzioni sono riportate in nero altre in grigio. Solo quelle in nero sono attivate e quindi utilizzabili. Dopo avere compiuto l'operazione sopra descritta apparirà sullo schermo una finestra) denominata 'A:/' indicante che si tratta di dati provenienti dal drive 'A' e priva di nome). La finestra) vedi anche Fig. 15) è costituita da una parte superiore retinata che serve per spostare la finestra con le dimensioni attuali.

Provate a puntare il cursore sulla parte tratteggiata in alto, e spostare il mouse tenendo premuto il tasto. Il quadratino in alto a sinistra serve per far sparire l'icona. Il quadratino a destra porta invece la finestra alle massime dimensioni. Quando una finestra viene chiusa bisogna ripetere

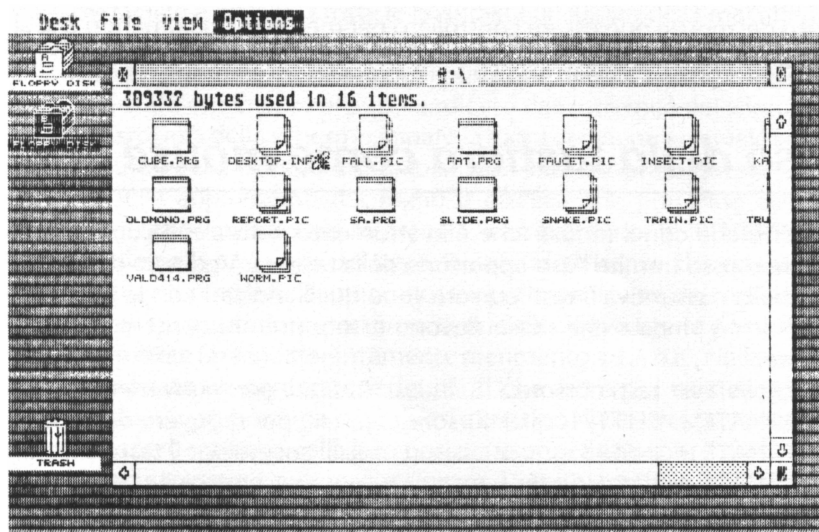



Fig. 15. Finestra con icone.

le operazioni descritte in precedenza per recuperarle. Il quadratino in basso a destra serve per modificare le dimensioni della finestra senza però mai riuscire a eliminarla completamente. Quando non tutte le informazioni possono essere viste nella finestra appaiono nella barra verticale destra e in quella orizzontale inferiore una parte retinata e una bianca. Quella bianca può essere mossa come un cursore per poter vedere nella finestra tutto il contenuto. Le frecce poste alle estremità effettuano la stessa operazione passo passo. Sotto alla barra indicante il nome della directory e del drive, è presente il numero di byte contenuti nei file presenti su disco e il loro numero. I file possono essere programmi, dati, linguaggi e altro ancora. È da segnalare come questi dati possono essere mostrati sotto forma di icone oppure di testo.

Riassumiamo nella tabella seguente il significato dei comandi.

Posizione simbolo	Funzione
Retinatura superiore	Muove finestra
Alto sinistra	Elimina finestra
Alto destra	Apri completamente finestra
Basso destra	Modifica dimensioni finestra
Barra bianca nel contorno	Scrolling vertic./orizzont.

File e uso del disk drive

Abbiamo gia accennato all'uso del drive nel capitolo precedente.

Vediamo ora in dettaglio come si trattano i file. Quando si vuole caricare da disco un programma è sufficiente inserirlo nel disk drive) sempre quando la luce rossa è spenta) e quindi premere il tasto del mouse sull'icona del floppy disk 'A') oppure indifferentemente 'B') . Passiamo ora al menù discedente 'File' e quindi possiamo scegliere l'opzione 'Open' oppure 'Show Info'.

OPEN .... apre il drive e permette il caricamento di un programma.

SHOW INFO fornisce informazioni sul disco inserito, il numero di byte liberi, eccetera.

Veniamo ora alla spiegazione di alcuni termini che ricorrono spesso nella terminologia informatica ma che spesso non si conoscono.

FILE (flusso in italiano) indica una sequenza di componenti del medesimo tipo , possono essere numeri, caratteri, o altro ancora. Un file viene chiuso da un elemento indicatore chiamato Eof) End Of File : fine del file) . Il file può essere anche un programma eseguibile diretta-

mente dal computer oppure tramite un linguaggio inserito in precedenza) es. BASIC o LOGO) .

DIRECTORY — è un indice del contenuto del disco. Ad essa può essere associato un nome per un più veloce rintraccio di file. Si tratta di un sistema ad albero in grado di avere ad ogni elemento della directory un'altra directory ad un livello inferiore.
es. disco1/programmi/giochi/scacchi
il programma 'scacchi' si trova nella directory 'giochi' che si trova nella directory 'programmi' che si trova nella directory 'disco1'.

Nello schermo appaiono alcuni tipi di icone : un documento con indicato lo spazio per il nome che rappresenta una directory, un rettangolo con una parte retinata rappresenta un programma eseguibile, una serie di fogli con un lembo sollevato rappresentano un file, un documento.

Nel menù VIEW sono presenti due opzioni:

SHOW AS ICONS — mostra i file sotto forma di icone con indicato il solo simbolo del documento e il nome.

SWOW AS TEXT — mostra i file indicando il nome del programma) le directory sono segnalate da un rettangolo con un rombo inserito, sulla sinistra del nome stesso) .
Segue quindi il suffisso inserito nel nome dopo il punto.
Es. BASIC.PRG indica che il file è un programma eseguibile denominato BASIC. SCACCHI.BAS indica che il programma denominato scacchi è scritto in BASIC.
I suffissi possono essere inseriti a piacere dall'utente facendo attenzione che solo quelli con il suffisso .BAS possono essere letti ed eseguiti da BASIC.
Dopo queste informazioni è indicato il numero di byte del file quindi la data e l'ora di realizzazione (vedi Fig. 16).

Un file) o una serie di file) possono essere copiati indicando l'icona interessata e quindi trascinandola in un altro punto dello schermo. Dopo questa scelta apparirà una scritta che chiede se si vuole realmente eseguire la copia del file. Viene indicato anche il numero totale di copie da effettuare. Durante la copia si verificherà un conflitto di nomi e sarà quindi necessario inserire quello nuovo. Si possono aprire contemporaneamente anche più finestre ma solo una alla volta sarà attiva) è indicata in nero) .

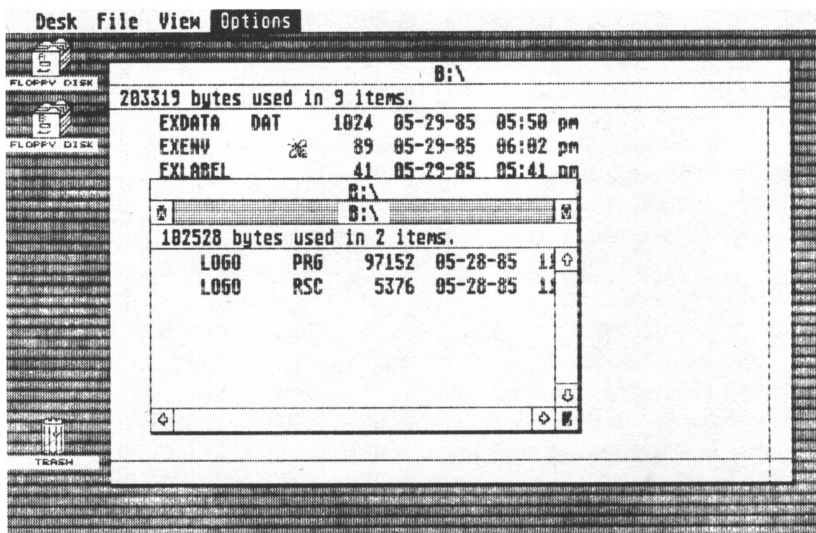



Fig. 16. Finestre con indicazioni in forma TEXT.

Messaggi di errore e di dialogo

Durante l'uso del computer possono apparire delle scritte all'interno di rettangoli al centro dello schermo. Questi messaggi sono di due tipi : messaggi di errore che vengono fatti comparire quando si verifica un evento non prestabilita e messaggi informativi. I primi generalmente offrono due possibilità quella di continuare oppure di ritornare ad una situazione precedente. la più tipica è quella di chiedere il caricamento di un file evitando di inserire il dischetto. Il computer comunicherà di controllare il disk drive, i collegamenti con l'eventuale hard disk ed infine propone due soluzioni. si può cancellare il comando) caricamento del file) oppure riprovare) naturalmente è neccessario inserire prima un disco) . Quando si presentano due opzioni) come in questo caso) uno dei due riquadri risulta essere bordato in grosso.

Copia di un disco

Quando si acquista un programma contenuto su un dischetto o più semplicemente quando si usano gli stessi dischi del sistema operativo forniti unitamente alla macchina, la prima cosa da fare è eseguire una copia ad uso STRETTAMENTE personale. È da sottolineare il fatto che la copia, detta anche di sicurezza, viene eseguita unicamente per cautelarsi contro involontari e accidentali cancellazioni del dischetto. La copia per altre persone e quindi lo smercio di programmi non di propria produzione è reato e come tale è punito. In questi ultimi anni si è fatto molto parlare di legislazioni sui diritti di autore nel software ma una legge organica e completa non è mai stata varata. Si rimanda quindi, oltre alle restrizioni legali, il lettore al suo buon senso e onestà. Affermare che il software dovrebbe essere gratuito per chi acquista una macchina, come qualcuno vorrebbe , sarebbe la stessa cosa di pretendere la benzina gratuitamente per il solo motivo di aver acquistato l'automobile. Dietro ai programmi più copiati) e quindi più utili ed efficaci) ci sono centinaia di ore di lavoro che devono forzatamente essere ricompensate. Dopo questa breve ma ritengo utile disquisizione sulla copia veniamo alla fase 'fisica della copia stessa'. Supponiamo di dover copiare un dischetto che per comodità chiamiamo 'A' e in cui per sicurezza abbiamo inserito la protezione. Prendiamo un dischetto vergine che chiameremo 'B' ed eseguiamo le seguenti istruzioni.

Il dischetto vergine deve essere preventivamente formattato. A tale scopo è neccessario introdurlo nel drive) dopo aver caricato il TOS) puntare con il mouse nell'icona FLOPPY DISK A) oppure B) e scegliere sempre con il mouse il menù a discesa denominato 'FILE' l'opzione 'Format ...'. Un messaggio vi avvertirà che l'operazione esegue la cancellazione del contenuto del dischetto. Se sieti sicuri date CLIC nel rettangolo 'OK'. Apparirà quindi un nuovo rettangolo di dialogo con proposta la singola faccia e la DISK LABEL. Se non avete particolari neccessità premere 'FORMAT' e darete così inizio all'operazione oppure inserire un nome di vostro piacimento per distinguere il disco dagli altri. Durante la formattazione stessa un indicatore analogico segnalerà lo stato di avanzamento dell'operazione. Al termine apparirà il numero di byte disponibili nel dischetto) di norma 357376) . Alla successiva richiesta di FORMAT rispondete con EXIT e tornerete alla fase iniziale di sistema. Veniamo ora alla fase di copia vera e propria. Sarà fatto riferimento all'uso di un solo drive (la configurazione più tipica) facendo attenzione al fatto che i possessori di due drive potranno eseguire le stesse operazioni in maniera più facilitata. Con il mouse è neccessario agganciare l'icona con il drive 'A' e trascinarla (tenendo schiacciato il tasto) fin sopra il drive B. Verrete immediatamnete avvertiti che il contenuto del disco B verrà completamente perduto. Premete OK e quindi inserite il disco A nel drive e scegliete COPY nel rettangolo di dialogo che apparirà successivamente. A questo punto il computer vi chiederà di inserire alternativa-

mente il dischetto A e B fino a quando l'operazione di copia non sarà conclusa. Questa operazione di 'travaso' delle informazioni è sottolineata dalla presenza di due indicatori analogici dello stato di riempimento.

È bene scrivere sul dischetto) meglio se in una etichetta adesiva) il nome del programma e tutte le altre informazioni neccessarie. Il dischetto a questo punto dovrebbe essere protetto spostando la linguetta atta a questo scopo. Il disco può essere quindi controllato inserendolo nel drive e attivando l'icona relativa al DRIVE A e chiedendo OPEN nel menù discendente FILE.

Menù

Veniamo ora ad esaminare le parti che compongono i menù discendenti.

Desk

DESKTOP INFO

Si tratta di una scritta informativa sul GEM, copyright, data di realizzazione eccetera.

VT52 EMULATOR

Simula un terminale denominato VT52 e quindi l'ATARI può essere usato come terminale intelligente con host remoti. Premendo il tasto <UNDO> si ritorna al GEM e con il tasto <HELP> si configura il terminale.

CONTROL PANEL

Attiva un pannello di controllo che permette la modificazione dei parametri fondamentali. La parte retinata in alto consente di muovere la finestra in tutto lo schermo. Il rettangolo in alto a sinistra elimina il control panel. Le tre barre a sinistra servono a controllare i colori fondamentali che vengono selezionati nella parte bassa. I controlli sono modificabili come dei normali cursori. Sotto alla barra informativa è riportata l'ora e la data. Tali dati non sono mantenuti aggiornati e vanno quindi modificati prima di ogni sessione di lavoro.

Per modificare la data basta portare il cursore sul rettangolo e quindi premere il tasto del mouse in modo che vada in reverse. Usando quindi la tastiera si possono modificare questi valori. I due cursori sottostanti stabiliscono il tempo in cui si deve tenere premuto un tasto (della tastiera) prima che inizi la ripetizione.

Il secondo cursore stabilisce la velocità di ripetizione dei tasti. I simboli sottostanti servono a cambiare il tempo minimo tra due pressioni del tasto del mouse. I simboli posti sotto indicano la situazione del campanello di allarme e del rumore che segnala l'avvenuta pressione dei tasti.
< CANCEL > riporta il pannello nelle condizioni precedenti alle modifiche.

SET RS232 CONFIG.

Stabilisce le caratteristiche della porta RS232.

BAUD RATE

È la velocità di trasmissione dei dati con possibilità di scelta tra 9600, 4800, 1200 e 300. I numeri esprimono il numero di bit trasmessi al secondo.

PARITY

Il controllo di parità serve a verificare la presenza di errori nella tramissione dei byte. L'ultimo bit viene usato per indicare se il numero di bit posti ad un valore alto (1) sono pari o dispari. In questo modo è possibile controllare se un bit è stato trasmesso errato e quindi viene richiesto nuovamente l'intero byte. Le possibilità offerte sono : NESSUNO, DISPARI, PARI.

DUPLEX Può essere singolo o completo. Stabilisce chi deve mandare l'eco dei caratteri sullo schermo di chi trasmette.

HALF DUPLEX (o eco locale) il trasmittente stesso invia i caratteri sia alla porta che al proprio monitor.

FULL DUPLEX (o eco remoto) il trasmittente invia i caratteri alla porta e il ricevente li rispedisce indietro dopo averli letti. In questo caso (più macchinoso da gestire) il trasmittente può verificare la correttezza dei dati che ha spedito.

BITS/CHAR

Stabilisce il numero di bit di cui è composta la parola.

La scelta è tra 5 e 8 bit.

XON/XOFF

Può accadere che mentre un terminale stia trasmettendo, l'altro si distragga... Capita infatti che il terminale ricevente sia impegnato in altre operazioni e possa perdere qualche dato. Si usano quindi dei protocolli di handshake (letteralmente 'stretta di mano'). Questi protocolli permetto al trasmittente di sapere se è ascoltato. Il cosidetto XON/XOFF consente al terminale ricevente di bloccare il trasmittente fino a quando non ha concluso le operazioni in corso. Vengono usati due caratteri di controllo per segnalare il blocco e lo sblocco del trasmittente.

INSTALL PRINTER

Stabilisce le caratteristiche della stampante da collegare al computer.

PRINTER TYPE

Si può scegliere tra la stampante a punti (DOT) oppure a margherita (DAISY).

COLOR

Stabilisce se la stampante è ad un solo colore (B/W) oppure a più colori (COLOR).

PIXEL/LINE

È il numero di pixel per ogni linea di stampa. La scelta è tra 960 e 1280.

QUALITY

Decide la qualità di stampa della printer.

PRINTER PORT

Viene deciso se la porta serve a collegare una stampante oppure un modem.

PAPER TYPE

La scelta sulla carta è tra foglio continuo (FEED) oppure foglio singolo (SINGLE).

File

OPEN

Questa opzione permette di vedere il contenuto di un disco e serve anche a selezionare un file (sia esso un programma oppure un insieme di dati). Questa possibilità è offerta solo quando stato posto in reverse uno dei due drive. Il comando può essere sostituito dalla pressione successiva del tasto del mouse.

SHOW INFO

Mostra le informazioni relative al disco inserito nel drive. Viene indicato il nome del drive, del disco, il numero di folder, di file, di byte usati e di quelli liberi. Premendo il tasto del mouse o RETURN si esce dall'opzione.

NEW FOLDER

I FOLDER sono dei sottoinsiemi di file che permettono una più facile gestione dei dati partendo dal principio di raggrupparli in una serie di

contenitori. Ad ogni folder deve essere associato un nome da voi scelto. In questo modo si possono inserire dati di fornitori, clienti eccetera, chiamando un folder CLIENTI, FORNITORI.GAS.

Il disco naturalmente non deve essere protetto.

CLOSE

Compie le operazioni inverse rispetto a quelle descritte in OPEN. Cancella dallo schermo anche le finestre.

CLOSE WINDOW

Chiude soltanto le finestre.

FORMAT

È l'operazione di preparazione del disco per accettare i dati. Questa operazione deve essere effettuata sul disco prima di essere usato per la prima volta e cancella eventualmente tutti i dati esistenti. Crea delle tracce per il posizionamento della testina del disco. Questa operazione non può essere compiuta prima in quanto Ogni computer formatta in maniera differente i propri dischi. Il disco per essere formattato non deve essere protetto. Al disco può essere assegnata una label. Durante l'operazione di formattazione il computer mostra lo stato di avanzamento di questa operazione con un indice scorrevole sullo schermo. Vengono quindi mostrati il numero di byte che si possono inserire (di solito 357376) e quindi richiesto se si desidera riformattare o uscire.

Wiew

Seleziona il modo in cui devono essere rappresentati i file e l'ordinamento degli stessi.

SHOW AS ICONS
SHOW AS TEXT

Mostra i file sotto forma di icone o di testo in quest'ultimo caso i dati presentati sono maggiori.

SORT BY NAME
SORT BY DATE
SORT BY SIZE
SORT BY TYPE

Ordina i file secondo l'ordine alfabetico, la data, la lunghezza e il tipo di programma.

Options

INSTALL DISK DRIVE

Permette di aggiungere un certo numero di disk drive inserendo l'identificatore (una lettera) e il nome.

Questa installazione può essere salvata su disco tramite l'opzione SAVE DESKTOP. Questa opzione può essere usata anche per cancellare una installazione precedentemente creata.

INSTALL APPLICATION

Questa opzione funziona solo dopo aver selezionato un file programma. Verrà mostrato il nome dell'applicazione, il tipo di documento e il tipo di applicazione. L'opzione GEM acquisisce i dati dal GEM, TOS e TOS-TAKES PARAMETERS se si tratta di dati non GEM. Tutte le modifiche devono essere salvate usando l'opzione SAVE DESKTOP.

SET PREFERENCES

Le prime due opzioni determinano se si desidera che venga chiesta la conferma prima di ogni operazione di cancellazione o di copia di un file. Viene quindi stabilita la risoluzione grafica del computer (bassa, media, alta).

Quando si usa un monitor bianco e nero si può usare solo l'alta risoluzione. Le modifiche possono essere salvate su disco come visto in precedenza.

SAVE DESKTOP

Salva su disco le modifiche effettuate. Il disco deve contenere il TOS e non deve essere protetto.

PRINT SCREEN

Esegue l'hard copy dello schermo del computer. Al calcolatore dovrà essere collegata una stampante grafica ed in condizioni di ON LINE. L'operazione richiede qualche minuto per il completamento.

3

ERRORI, GUASTI E MANUTENZIONE

Per il corretto funzionamento del computer e delle sue periferiche bisogna avere qualche attenzione. Riassumiamo di seguito le cose che non si possono fare. Abbiamo gia avuto modo di dire che la superficie sulla quale deve essere fatto scorrere il mouse deve essere liscia e pulita. In particolar modo non deve essere umida o ancora peggio bagnata. I sensori posizionali sono particolarmente sensibili e possono deteriorarsi a contatto con l'acqua o con altri liquidi. È abbastanza inutile ribadire che il piano di scorrimento non deve essere inclinato (il mouse potrebbe scivolare lentamente a terra ...). Il computer e nemmeno le periferiche devono essere coperti o sovrastati da libri, fogli o altro. Le fessure di aerazione devono permettere lo smaltimento del calore prodotto internamente. Non si devono sottoporre a fonti di calore o magnetiche i dischetti in quanto risultano particolarmente sensibili a tali influssi. Le connessioni che si trovano nella parte posteriore del computer o quelle che collegano le periferiche non devono essere toccate con le mani o con oggetti mettalici. Oltre a queste raccomandazioni bisogna ricordare che il computer non deve essere sottoposto ad alcuna sollecitazione meccanica specie durante il trasporto. Per quanto riguarda il drive bisogna tenere presente che il disco non deve essere inserito o estratto quando la luuce del drive stesso risulta accesa. Il computer e le sue periferiche non hanno bisogno di alcuna operazione di manutenzione ad esclusione del mouse. Muovendosi infatti su un piano potrà sporcarsi la sfera di gomma. È sufficiente estrarla facendo scorrere la protezione sotto al mouse e quindi prelevare la sfera e lavarla con acqua e sapone. Non devono essere usati assolutamente solventi o sostanze che possono deteriorare la gomma. Le operazioni sono illustrate in Fig. 17 e 18.

Naturalmente la sfera dopo la pulizia deve essere accuratamente asciugata e delicatamente inserita nell'alloggiamento. È importante non forzare i riferimenti posizionali all'interno del mouse perchè sono molto delicati.

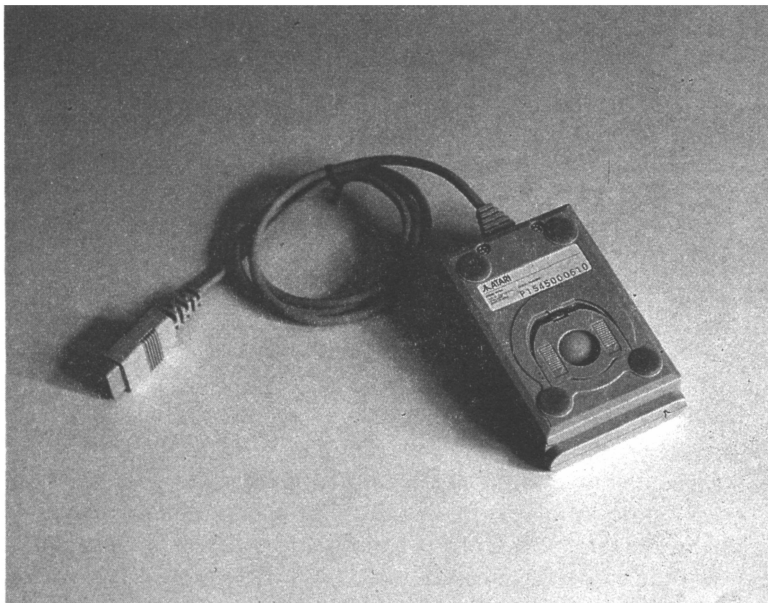



Fig. 17. Apertura del mouse.

Fig. 18. Estrazione della sfera posizionale.

I LINGUAGGI DI PROGRAMMAZIONE

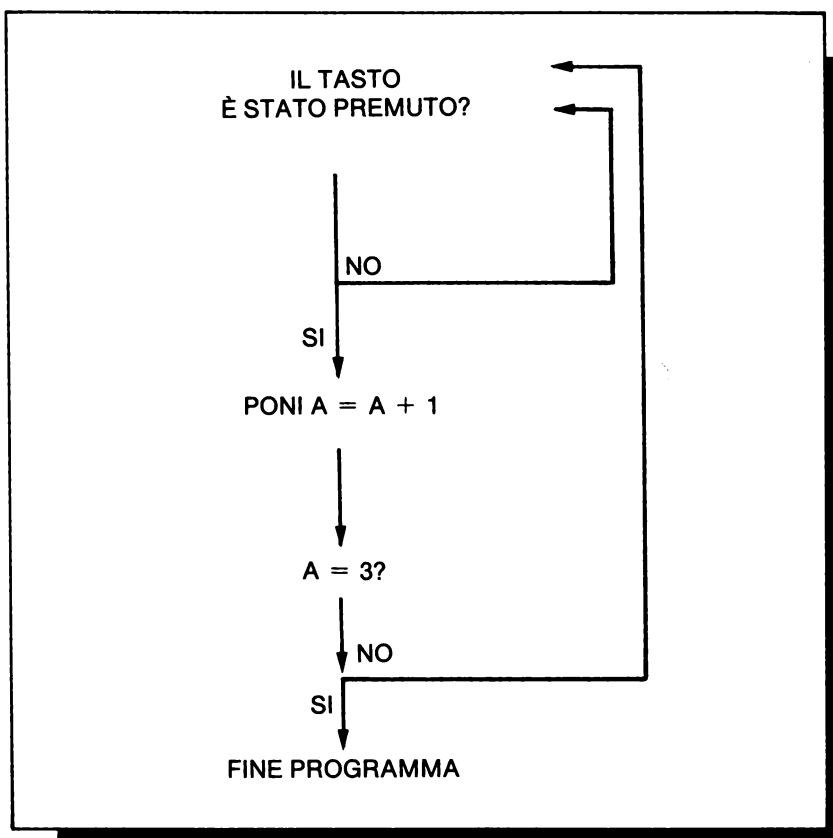
Il computer tratta tutte le informazioni trasformandole in numeri binari, composti di 1 e 0 e corrispondenti alle condizioni di ON/OFF di un interruttore. Le informazioni da inserire dovrebbero essere composte da questi numeri. Si tratta come si vede di un'assurdità data la complessità delle operazioni da compiere e del loro numero anche per ottenere risultati elementari. Sono stati, per questo motivo, creati dei linguaggi di programmazione che maggiormente si avvicinano a quello naturale dell'uomo. Questi linguaggi sono definiti 'ad alto livello' in quanto una sola istruzione serve ad eseguire un numero enorme di operazioni elementari. I linguaggi di programmazione sono numerosi e si differenziano per le caratteristiche delle macchine in cui sono stati implementati, per le loro caratteristiche particolari (esistono linguaggi scientifici, finanziari, orientati alla grafica, ...). I linguaggi maggiormente utilizzati sono il BASIC, il LOGO, il PASCAL, FORTRAN, LISP, COBOL eccetera.

5

LA PROGRAMMAZIONE

Vediamo ora come si imposta la programmazione di un computer.

Durante il nostro discorso faremo riferimento al linguaggio BASIC in quanto è il più diffuso. La prima cosa da fare è da individuare come scaturisce il risultato del problema che si intende risolvere. Si devono quindi ricercare i dati di partenza e quindi l'algoritmo (il metodo usato per raggiungere lo scopo) più adatto. I dati di partenza devono essere quindi trasformati in variabili e su di esse compiere le operazioni. Un valido aiuto alla programmazione consiste nei cosidetti DIAGRAMMI DI FLUSSO o DIAGRAMMI A BLOCCHI. Questi sono costituiti dal tracciato che si deve seguire per giungere al risultato. Si tratta di sostituire alle operazioni, dei blocchi contenenti il nome dell' operazione stessa. Si passa sucessivamente ad esaminare i singoli blocchi e si scompongono nuovamente. L'importante è comunque dividere il problema in una serie successiva di semplificazioni fino ad arrivare ad una posizione facilmente risolvibile. Durante la stesura dei diagrammi di flusso è necessario porre particolare attenzione alle diramazioni ovvero quando un programma può prendere due (o più) strade differenti a seconda di una serie di eventi che possono accadere. Un caso banale ma efficace per comprendere questo tipo di istruzioni è il seguente. Vogliamo creare un contatore in grado di terminare il proprio lavoro quando raggiunge un certo numero. Dobbiamo fare in modo che il computer possa essere in grado di riconoscere la pressione di un tasto (questa istruzione è presente in quasi tutti i linguaggi) e quindi stabilire ad esempio che il numero massimo sia 3.

Vediamo, nella pagina seguente, con uno schema, questo tipo di programma.

In BASIC il programma si sarebbe scritto :

```
10 INPUT A$
20 A=A+1
30 IF A=3 THEN 50
40 GOTO 10
50 END
```

Si rimanda comunque il lettore ad un testo specializzato per acquisire i rudimenti dell'arte programmatoria e in questo campo questa casa editrice è senz'altro la più fornita e assicura con delle firme prestigiose dei veri e propri corsi. Nel prossimo capitolo ci interesseremo comunque in maniera approfondita della programmazione del linguaggio BASIC che è di gran lunga il più diffuso e utilizzabile per la maggior parte delle applicazioni siano esse di tipo commerciale, scientifico o didattico.

6

IL LINGUAGGIO BASIC

Introduzione

Questo linguaggio è sicuramente il più utilizzato e diffuso nelle sue molteplici versioni e dialetti. BASIC è costituito dalle iniziali della frase Beginners All-purpose Symbolic Instruction Code (codice simbolico generale di istruzioni per principianti). Fu realizzato nel 1964 presso l'Università di Dartmouth di New Hampshire negli Stati Uniti. Quella che tratteremo ora può essere considerato come una versione standard utilizzabile per la maggor parte dei computer e quindi non solo per gli ATARI 520 ST e 1040 ST.

Uso del linguaggio

Un programma BASIC è composto da una serie di istruzioni precedute da un numero progressivo (numero di linea) che le identifica e che normalmente viene numerato di 10 in 10 in modo da poter eventualmente inserire altre istruzioni. Di solito una delle prime linee è costituita da un commento (in inglese REM) che contiene il nome del programma. Al fine di facilitare la lettura dei listati anche dopo molto tempo dalla stesura.

Esso appare solo nel programma e su di esso non agisce in alcun modo.

```
Es. 10 REM
```

Il BASIC posside due tipi di variabili, quelle numeriche e quelle alfanumeriche (o stringhe). Le variabili numeriche avranno come nome una combinazione qualsiasi di lettere e numeri.

Ad es. A, DE5RE , PIGRECO eccetera. sono tutti nomi corretti; i seguenti invece non saranno accettati:

```
5gt , gt.hg , hg/?tt, ...
```

Non dovranno cioè apparire caratteri diversi dalle lettere e numeri e questi ultimi non dovranno essere come carattere iniziale.

I nomi delle variabili alfanumeriche saranno composte come quelle numeriche ma avranno come suffisso il carattere '$'.

```
Es. as$ , urt$, ...
```

Non si possono usare come nome di variabili le parole riservate ovvero le parole chiave (ad esempio PRINT, REM, eccetera).

Il BASIC possiede delle istruzioni che possono essere usate in maniera diretta e indiretta. Le istruzioni dirette sono quelle che vengono eseguite subito, non hanno numero di linea e non vengono 'ricordatè dal computer.

```
Es. PRINT 5
```

Stampa sullo schermo il numero 5 ma dopo di questo l'istruzione non può più essere riutilizzata. Vediamo ora alcune semplici istruzioni.

PRINT Stampa sullo schermo quello che viene inserito dopo. Può anche eseguire un serie di calcoli.

```
Es. PRINT 5+3
```

Avremo come risultato il numero 8. Si possono usare come operatori (+ – * /) che significano rispettivamente somma, sottrazione, prodotto e divisione.

```
Es. PRINT "qwerty"
```

LET — Stampa la stringa 'qwerty' sullo schermo (apici esclusi).
Assegna un valore ad una variabile sia essa numerica o alfanumerica.

```
Es. LET a=4
```

Pone al valore 4 la variabile denominata 'a'.

```
Es. LET d=a+6/r
```

Pone la variabile 'd' uguale al risultato dell'espressione.

```
Es. LET a$="Atari 520 st"
```

INPUT — Pone la variabile a$ uguale al contenuto tra gli apici.
Accetta dall'esterno il valore di una variabile (numerica o alfanumerica). Ad esempio:

```
20 INPUT a
```

Attende che l'utente inserisca un numero che verrà assegnato alla variabile 'a'.

```
20 INPUT a$
```

GOTO — Attende una stringa seguita da RETURN.
Salta ad una linea indicata di seguito.

```
Es. 10 PRINT "a"
    20 GOTO 10
```

Il programma va 'in loop' cioè continua a girare senza mai fermarsi. L'unico modo di bloccare il programma è quello di premere contemporaneamente i tasti CONTROL + G (solo per ATARI 520 ST).

GOSUB e RETURN Come per il precedente ma in questo caso si fa riferimento ad un salto con il ritorno all'istruzione successiva a quella del GOSUB.

```
Es.   10 REM PROVAGOSUB
      20 PRINT "QWERTY"
      30 GOSUB 1000
      40 PRINT "ASDFG"
      50 GOSUB 1000
      60 END
    1000 REM SUBROUTINE
    1010 PRINT "SUB"
    1020 RETURN
```

Avremmo come risultato:

```
QWERTY
SUB
ASDFG
SUB
```

END Termina un programma.

FOR .. TO e NEXT Crea un ciclo che vine ripetuto n volte secondo questo schema:

```
FOR I=1 TO n
< istruzioni da eseguire >
NEXT I
```

Facciamo un esempio:

```
10 FOR I=4 TO 7
20 PRINT I
30 NEXT
```

avremmo come risultato:

```
4
5
6
7
```

È possibile stabilire anche il passo del ciclo eventualmente anche negativo.

```
10 FOR I=1 TO 3 STEP .5
20 PRINT I
30 NEXT I
```

Avremmo come risultato:

```
1
1.5
2
2.5
3
```

Altro caso con i numeri negativi:

```
10 FOR A=10 TO 5 STEP -1
20 PRINT A
30 NEXT A
```

il risultato sarà:

```
10
 9
 8
 7
 6
 5
```

IF .. THEN — Esegue una istruzione solo quando si verifica una certa condizione.

```
Es. 10 PRINT "DAMMI UN
      NUMERO"
    20 INPUT A
    30 IF A=0 THEN PRINT
      "ZERO"
    40 END
```

Il programma scrive "ZERO" solo quando si inserisce in input '0'.

RUN — Manda in esecuzione un programma.

NEW — Cancella un programma dalla memoria.

Le finestre

Dopo aver caricato da disco il BASIC appariranno sullo schermo una serie di finestre.

LIST
La finestra visualizza le istruzioni del programma. Può essere attivata con l'omonimo comando inserito con la tastiera o tramite il mouse (vedi paragrafo 6.4).

OUTPUT
Visualizza il risultato dell'elaborazione sia essa grafica che alfanumerica.

COMMAND
Permette l'inserimento di un programma o la stesura di istruzioni dirette.

EDIT
Permette la correzione , modificazione del programma. Può essere ottenuta scrivendo EDIT (o ED) in modo COMMAND oppure usando la relativa istruzione nel menù discendente.

Le finestre possono essere ingrandite, spostate nella loro posizione, eliminate eccetera. È possibile anche uno scroll (scorrimento) sia esso verticale che orizzontale.

Menù

Di questo menù si è già parlato in precedenza e quindi si rimanda a quelle righe il lettore.

FILE
Gestisce le fasi di INPUT/OUTPUT da disco.

LOAD
Carica un programma. Deve essere inserito un disco nel drive e apparirà dopo qualche secondo l'elenco dei programmi contenuti. Sarà

sufficiente puntare su essi il mouse e confermare (o premere velocemente due volte il tasto del mouse sul nome stesso) per effettuare il caricamento. Il nome del programma può essere digitato da tastiera.

SAVE AS
Salva un programma su disco. Le modalità sono identiche all'opzione precedente. Se si inserisce un nome esistente il programma memorizzato su disco viene perso a favore di quello nuovo.

DELETE FILE
Cancella un programma. Il metodo usato è quello acquisito in precedenza.

MERGE
Fonde un programma esistente su disco con un'altro che è presente in memoria. Le istruzioni che hanno lo stesso numero nei due programmi entrano in conflitto e prevale quello caricato da disco.

QUIT
Esce dal BASIC (si perdono eventuali programmi in memoria).

RUN

RUN
Manda in esecuzione un programma.

BREAK
Interrompe momentaneamente l'esecuzione di un programma.

STOP
Blocca definitivamente un programma.

CONTINUE
Fa continuare un programma interrotto con BREAK.

STEP
Ogni istruzione viene eseguita solo dopo che è stato premuto <RETURN>. Viene usato per la messa a punto dei programmi.

BUF GRAPHICS
Inserisce e disinserisce la memoria video. Quando è disinserita una immagine viene persa dalla finestra OUTPUT quando questa viene spostata o vengono eseguite operazioni su di essa. L'attivazione (o la disattivazione del comando) comporta la perdita del programma esistente in memoria. Quando viene disinserita la memoria di schermo si ha a disposizione un maggiore spazio per i programmi.

EDIT

START EDIT

Apre la finestra EDIT che permette la compilazione, la modifica e la cancellazione delle linee di programma.

EXIT EDIT

È un comando attivo solo quando è aperta la finestra EDIT. Ritorna alla fase di COMMAND.

HELP EDIT

Mostra l'elenco dei tasti funzione. Vedi anche paragrafo 6.10.

GOTO LINE ...

Funzionante solo in EDIT come la maggior parte dei comandi che seguono. Sposta la finestra di edit al numero di linea indicato nella finestra che appare appena si fa eseguire il comando.

DELETE LINES

Cancella le linee comprese tra le due estreme indicate nell'apposito 'dialog box'.

INSERT SPACE

Inserisce uno spazio prima della posizione del cursore.

DELETE CHAR

Cancella il carattere indicato in questo momento dal cursore.

INSERT LINE

Inserisce una linea vuota prima della posizione del cursore.

REMOVE LINE

Cancella la linea puntata dal cursore.

PAGE UP

Fa lo scroll verso l'alto di una pagina.

PAGE DOWN

Come per la precedente ma questa volta è verso il basso.

LOAD TEXT

Carica da disco un file salvato con il comando SAVE TEXT.

SAVE TEXT

Salva su disco (senza inserire il nome) lo schermo dell'EDIT. Quando viene richiamato il testo è in grigio e deve quindi essere confermato con il tasto di RETURN.

NEW BUFFER

Cancella il file richiamato in precedenza con il comando LOAD TEXT.

LIST

Esegue la lista del programma sulla finestra che ha lo stesso nome.

DEBUG

TRON

È la diminuzione di TRACE ON e offre quindi la 'traccia del programma' durante l'esecuzione. Si tratta di una serie di comandi molto utili nella fase di messa a punto del software (DEBUG letteralmente spulciare). Fornisce la sequenza delle linee di programma eseguite. È possibile anche decidere quali sono le linee che devono essere sottoposte alla fase di TRACE.

TROFF

Interrompe l'esecuzione del comando precedente.

TRACE

Simile a TRON ma questa volta viene visualizzato anche il contenuto della linea di programma e non solo il suo numero.

UNTRACE

Esclude l'esecuzione di TRACE.

STESURA DI UN PROGRAMMA

Veniamo ad analizzare ora alcune caratteristiche particolari che riguardano principalmente i computer ATARI 520 ST e 1040 ST.

Variabili

Delle variabili in generale abbiamo avuto modo di parlare in precedenza. Analizzeremo ora solo alcuni aspetti particolari. Le Variabili alfanumeriche possono essere assegnate anche componendo più parti di stringa tramite l'operatore '+'.

Vediamo ora questo esempio:

```
10 LET a$="aaa"+"bbb"
20 PRINT a$
```

Avremmo come risultato dell'elaborazione la stringa aaabbb.

La stessa operazione può essere fatta assegnando a diverse variabili le relative stringhe e poi facendone la somma.

```
Es. 10 a$=??qqq??
    20 b$=??eee??
    30 c$=??ttt??
    40 q$=a$+c$+b$
    50 PRINT q$
```

risultato:

```
qqqtttee e
```

Come si vede dall'esempio le variabili possono essere assegnate anche rinunciando all'istruzione LET. Basterà porre il nome della variabile uguale al valore che vogliamo assegnare.

Le variabili possono essere date anche sotto forma di matrici.

In questo caso bisognerà dimensionare la matrice stessa (necessario solo se il numero di elementi che si intendono usare è maggiore di 10). L'istruzione da utilizzare è DIM seguito dal nome della variabile e (fra prentesi tonde) dal numero di elementi.

Es. DIM a(n) dimensiona una matrice denominata 'a' e composta di 'n' elementi.

Le matrici possono quindi essere assegnate come delle normali variabili ricordandosi di aggiungere ad esse sempre il relativo numero di ordine.

```
Es. 10 DIM a(12)
    20 a(4)=6
    30 a(11)=a(4)
    40 PRINT a(11)

    RUN

    6
```

Le matrici possono essere anche alfanumeriche in questo caso bisognerà segnalare al computer la circostanza aggiungendo il simbolo del dollaro come suffisso.

```
Es. 10 DIM a$(12)
    20 a$(4)= "test"
    30 a$(11)=a$(4)
    40 PRINT a$(11)

    RUN

    test
```

Le variabili possono essere intere o reali. Quelle reali possono essere suddivise ancora a semplice o a doppia precisione. Le variabili intere sono seguite dal simbolo della percentuale.

```
Es. 10 A% = 3
    20 PRINT A
    RUN

    3
```

Altro esempio:

```
10 A = 234/17
20 B% = 234/17
30 PRINT "A = ";A
40 PRINT "B% = ";B%

RUN

13.7647
13
```

La grafica

È senz'altro una delle caratteristiche più interessanti di questo computer in quanto la gestione è facile ed a alta risoluzione. Si può usare il computer unitamente ad un monitor monocromatico (ad alta risoluzione) oppure con un monitor a colori (risoluzione media e bassa).

I modi sono i seguenti:

RISOLUZIONE	pixel X	pixel y	colori
alta	640	400	2
media	640	200	4
bassa	320	200	16

Ad ogni colore è assegnato un codice numerico secondo la seguente tabella.

COLORE	N.	risoluzione
BIANCO	0	A M B
NERO	1	A M B
ROSSO	2	M B
VERDE	3	M B
BLU	4	B
TURCHESE	5	B
GIALLO	6	B
MAGENTA	7	B
BIANCO	8	B
NERO	9	B
ROSSO CHIARO	10	B
VERDE CHIARO	11	B
BLU CHIARO	12	B
CELESTE CHIARO	13	B
GIALLO CHIARO	14	B
MAGENTA CHIARO	15	B
dove B: BASSA RISOLUZIONE M: MEDIA RISOLUZIONE A: ALTA RISOLUZIONE		

Il computer crea i disegni (ma anche i caratteri dei testi) ricorrendo alla modificazione del colore dei singoli pixel. Il pixel è la più piccola unità indirizzabile sullo schermo.

Significa che ogni pixel altro non è che un puntino che può risultare acceso o spento. Se è acceso (detto anche INK color: colore dell' inchiostro) avrà un colore differente rispetto a quello di fondo (detto anche PAPER: carta).

La grafica dell'ATARI fa riferimento ad un sistema di assi cartesiani che trova la sua origine in alto a sinistra dello schermo.

Suono

Il computer è in grado di realizzare suoni e rumori grazie ad un processore dedicato. Vediamo come può essere utilizzato. Il comando BASIC relativo è SOUND che viene usato in questo modo:

```
SOUND a,b,c,d,e
```

Dove **a**: è la voce (tra 1 e 3 , il 4 indica il rumore)
b: è il volume
c: è la nota
d: è l'ottava
e: è la durata

Provate il seguente esempio:

```
10 FOR I=1 TO 15
20 FOR J=1 TO 15
30 SOUND 1,8,I,J,9
40 NEXT J
50 NEXT I
```

Per un uso più specialistico dell'uscita audio rimandiamo il lettore a testi specializzati in quanto la gestione dei suoni e dei rumori richiede conoscenze non solo di musica ma anche (e specialmente) delle caratteristiche del processore.

La stampante

La stampante è uno dei componenti più utili del sistema in quanto permette di trasferire su carta (e quindi in maniera facilmente consultabile anche lontano dalla macchina) i dati elaborati dal computer. Le stampanti possono essere collegate tramite il connettore posteriore del computer e vengono azionate da una serie di comandi speciali. Vediamo i più importanti. È possibile eseguire la copia dello schermo (almeno nelle versioni provate) premendo contemporaneamente i tasti ALTERNATE e HELP disponendo di una stampante grafica.

LPRINT come per il comando PRINT ma questa volta l'uscita avverrà sulla stampante e non sullo schermo.
LLIST esegue la lista delle istruzioni del programma memorizzato nel computer.

È possibile sfruttare al massimo le caratteristiche della stampante posseduta (caratteri speciali, in grassetto, sottolineato, ecc.) inviando alla stessa una serie di codici di controllo tramite LPRINT seguendo le istruzioni che sono contenute nei manuali della stampante.

Condizioni (AND,OR,NOT..)

Quando si usano strutture sintattiche del tipo IF certa condizione THEN .... è neccessario ricorrere ad alcuni operatori booleani che ci permettono di definire situazioni che si possono riassumere in questo modo. Una prima considerazione riguarda l'algebra di Boole. Si possono avere solo due condizioni 1 0 (acceso / spento o vero / falso). Queste condizioni corrispondono a livello di circuiti elettronici ad una situazione di segnale basso o alto. Supponiamo di avere un circuito con due ingressi ed una sola uscita.

Se applichiamo un segnale alto (d'ora in avanti = 1) all'ingresso 1 e 2 avremo un segnale alto in uscita , basso in tutte le alte combinazioni di ingresso secondo questa tabella.

OPERATORE AND	IN 1	IN 2	OUT
	0	0	0
	0	1	0
	1	0	0
	1	1	1

In BASIC possiamo usare la stessa tabella sostituendo mentalmente la condizione 1 a 'verò e 0 a 'falso'.

```
Es. 10 a=3
    20 b=4
    30 IF a=3 AND b=4 THEN
       PRINT "a=3/b=4"
    40 END
```

Vediamo che se le due condizioni sono entrambe vere il programma stampa quello che appare tra apici altrimenti termina.

Di seguito diamo gli altri operatori booleani facendo attenzione al modo in cui compaiono nelle frasi BASIC.

OPERATORE OR	IN 1	IN 2	OUT
	0	0	0
	0	1	1
	1	0	1
	1	1	1

```
Es. 10 a=2
    20 b=5
    30 IF a=2 OR b=2 THEN
       PRINT "a/b = 2"
    40 END
```

OPERATORE NOT	IN 1	OUT
	0	1
	1	0

Possiamo usare gli operatori logici anche nel seguente modo:

```
10 a=2
20 b=5
30 PRINT (a=2) and (b=5)

RUN
```

Se le condizioni sono vere (come nell'esempio) avremmo come risultato -1 , zero in caso contrario. Provate a cambiare i valori alle linee 10 e 20 ed otterrete risultati sempre falsi.

Questo tipo di operazione può essere fatta con tutti gli operatori logici ed eventualmente anche con più di essi contemporaneamente.

Condizioni di errore

Nel linguaggio BASIC è possibile che si verifichino situazioni di errore che portano al blocco del programma. È possibile comunque evitarlo con delle particolari istruzioni che vedremo di seguito. La più importante è ON ERROR GOTO .. con la quale si manda il programma ad una determinata linea in caso si verichi un errore. Si può quindi controllare il tipo di errore in base al codice che lo contradistingue. Un esempio può facilitare molto la comprensione di questo argomento. Rimando il lettore ad un attento esame dei listati di programmi pubblicati in questo libro e nei quali appaiono controlli di errore in maniera intensiva.

```
Es. 10 ON ERROR GOTO 100
    20 PRINT 3/0: REM (errore
       n.11 divisione per zero)
    30 PRINT "fine programma"
    40 END 100 IF ERR = 11 THEN
       PRINT "divisione per zero"
   110 RESUME 30
   120 GOTO 30
```

La variabile di sistema ERR contiene il numero di errore come è riportato nel paragrafo 6.8 di questo testo. RESUME dovrà indicare il numero di linea da cui si vuole continuare il programma.

Uso del drive

Abbiamo gia avuto modo di vedere l'uso del drive per memorizzare i programmi BASIC. Il disco può essere vantaggiosamente usato anche per memorizzare dati e informazioni. In questa ottica risulta uno strumento indispensabile per la creazione di archivi di grandi quantità di dati siano essi numerici o alfanumerici.

I dati possono essere raggiunti (e quindi anche memorizzati) in forma sequenziale o casuale. In modo sequenziale i dati sono uno di seguito all'altro e devono essere letti tutti quelli che vengono prima per raggiungere il valore che ci interessa. Se ad esempio vogliamo raggiungere il dato 'n' dovremmo leggere tutti gli n valori e trattenerci solo l'ultimo. Nel caso di dati casualmente memorizzati un indice (o un codice) ci permette di raggiungerli immediatamente. Vediamo ora due programmi che possono spiegare meglio il funzionamento del metodo.

Prima di utilizzare il programma inserire nel drive un disco non protetto e formattato sul quale andranno memorizzati i dati.

```
 10 REM file sequenziale
 20 n = 5: REM numero massimo
    di dati
 30 FOR i = 1 TO n
 40 INPUT "Dammi il numero";a(i)
 50 NEXT i
 60 a$ = "sequenz": REM nome
    del file
 70 OPEN "O",1,a$
 80 FOR i = 1 TO n
 90 PRINT 1,a(i)
100 NEXT
110 CLOSE 1
```

Con questa parte di programma abbiamo memorizzato su disco un file sequenziale che conterrà di seguito i dati inseriti (nell'esempio 5).

```
120 INPUT "Dammi il numero del
    dato da cercare";nr
130 IF nr > n THEN 120
140 OPEN "I",1,a$
150 FOR i = 1 TO nr
160 INPUT 1,w
170 NEXT i
180 CLOSE 1
190 PRINT "Valore cercato =";w
200 GOTO 120
```

In questo modo viene fatto un ciclo da 1 al numero del dato da cercare e quindi letti i valori nel file e stampato solo quello di posizione nr.

Nel caso di file ad accesso casuale avremmo ogni dato ben identificato e quindi solo quello viene caricato.

```
 10 REM file ad accesso casuale
 20 n = 5: REM numero massimo
    di dati
 30 FOR i = 1 TO n
 40 INPUT "Dammi il numero";a(i)
 50 NEXT i
 60 FOR i = 1 TO n
 70 a$ = "Q" + CHR$(i + 64):REM
    nome del dato
 80 OPEN "O",1,a$
 90 PRINT 1,a(i)
100 CLOSE 1
110 NEXT i
```

In questo modo vengono creati tanti file quanti sono i dati e contengono ognuno un solo valore. È stata usata la funzione CHR$ che trasforma una variabile numerica nel relativo carattere secondo il codice ASCII.

```
120 INPUT "Dammi il numero del
    dato da cercare"; nr
130 IF nr > n THEN 120
140 A$ = "Q" + CHR$ (nr + 64)
150 OPEN "I", #1, a$
160 INPUT #1, w
170 CLOSE #1
180 PRINT "Valore cercato
    ="; w
190 GOTO 120
```

In questo modo dovrebbe risultare chiara la differenza tra i due metodi ma si pone il problema: quale dei due è migliore ? Anche in questo caso la verità sta in mezzo: dipende dall'uso che se ne deve fare del programma e da quanti dati è neccessario avere a disposizione. Se la mole delle informazioni è notevole è bene raggruppare tutti i dati in file che vengono ricercati da altri puntatori come avviene in una struttura ad albero. Il metodo dei file sequenziali permette di ricercare un dato direttamente da nome e quindi di più facile utilizzo ma è bene comunque ricorrere sempre a una routine di errore per evitare di cercare per sbaglio un file che non esiste.

Elenco istruzioni Basic

Di seguito sono indicate le istruzioni BASIC con la forma ed alcuni esempi di utilizzo. Le istruzioni, i comandi e le funzioni sono scritti in ordine alfabetico.

ABS

Formato: ABS(n)

Restituisce il valore assoluto di n.

```
Es. 10 PRINT ABS(102)
    RUN
    102
```

```
Es. 10 PRINT ABS(-102)
    RUN
    102
```

ASC

formato: ASC(a$)

Ritorna un intero compreso tra 0 e 255 corrispondente al codice ASCII del primo carattere della stringa A$. Se la stringa è nulla si ha errore.

```
Es. 10 PRINT ASC("b")
    RUN
    98
```

ATN

formato: ATN(n)

Calcola l'arcotangente del numero 'n'. N dovrà essere compreso tra –pigreco/2 e +pigreco/2.

```
Es. 10 PRINT ATN(0.157)
       RUN
       0.155728
```

AUTO

formato: AUTO
AUTO a
AUTO a,b
AUTO .b

Dopo il comando AUTO la numerazione inizia con il numero indicato da 'a' e si incrementa di 'b'. 'a' e 'b' dovranno essere interi positivi . Il numero di linea dovrà essere compreso tra 0 e 65535. La numerazione automatica viene interrotta premendo contemporaneamente i tasti CONTROL e G.

Se il numero di linea è gia esistente appariranno davanti due asterischi.

Es. AUTO
genera 10,20,30 ...

Es. AUTO 100
genera 100,110,120 ...

Es. AUTO 10,2
genera 10,12,14 ...

B

BLOAD

formato: BLOAD a$,l

Carica un file contenente un programma in linguaggio macchina all'interno della memoria. Il file ha il nome a$ e viene caricato dall'indirizzo 'l'. L'indirizzo dovrà avere un valore compreso tra 0 e 65535. Bload può salvare anche immagini su disco.

BREAK

formato: BREAK
BREAK n
BREAK nome label

Ferma l'esecuzione di un programma ad una linea o ad una label . Il comando CONT o RETURN fa eseguire dalla prossima linea. Per uscire dal BREAK scrivere STOP o END.

```
Es. 10 REM a
    20 REM b
    30 REM c
    40 REM d
    BREAK 30
    RUN
    b 30 REM c
```

BSAVE

formato: BSAVE a$,a,b

Salva una parte di memoria su disco con il nome a$, a partire dall'indirizzo 'a' e di lunghezza 'b'. Vedi anche BLOAD.

CALL

formato: CALL v
CALL v,p1,p2

dove 'v' è una variabile numerica, p1 e p2 sono due parametri.

Trasferisce il controllo ad una subroutine in linguaggio macchina. La variabile indica l'indirizzo di partenza, i parametri opzionali servono a trasferire i dati dal programma BASIC a quello in linguaggio macchina.

I parametri sono separati da una virgola.

CDBL

formato: CDBL (n)

Converte un numero di qualsiasi tipo in un reale a doppia precisione.

```
Es. 10 PRINT 4/3
    20 PRINT CDBL (4/3)

    RUN

    1.33333
    1.33333336
```

CHAIN

formato: CHAIN nomeprogramma,numero linee,ALL

CHAIN MERGE nomeprogramma,numero linee,DELETE a-b

Trasferisce il controllo e passa le variabili ad un'altro programma. Se si omette il comando ALL bisognerà indicare con COMMON le variabili da mantenere e trasferire.

CHR$

formato: CHR$(n)

Ritorna un carattere equivalente al codice ASCII corrispondente. 'n' dovrà essere intero e compreso tra 0 e 255.

```
Es. 10 PRINT CHR$(65)
    RUN
    A
```

```
Es. 10 FOR i=32 to 255
    20 PRINT CHR$(i);
    30 NEXT
    RUN
```

appaiono tutti i codici ASCII

CINT

formato: CINT (i)

Arrotonda un numero reale (n) all'intero più vicino. Il numero (n) deve essere compreso tra -32768 e +32767.

```
Es. 10 k=178/9.2
    20 a%=CINT(k)
    30 PRINT k
    40 PRINT a%
    RUN
    19.3478
    19
```

CIRCLE

formato: CIRCLE x,y,r,a,b

Disegna un cerchio alle coordinate x,y di raggio r con angolo iniziale a e finale b. 'a' e 'b' si esprimono in decimi di grado. Es. 37,1 gradi = 371).

Secondo questo schema.

```
Es. 10 CIRCLE 200,200,70,420,3115
```

CLEAR

formato: CLEAR
CLEAR,n

Cancella le variabili liberando la massima dimensione della memoria. Le variabili numeriche sono riportate a 0. Le variabili alfanumeriche sono riportate alla stringa nulla.

Non viene cancellato il programma esistente.

CLEARW

formato: CLEARW n

Pulisce la finestra 'n' che dovrà essere compresa tra 1 e 4 secondo questa tabella.

n.	finestra
1	LIST
2	OUTPUT
3	COMMAND
4	EDIT

```
Es. 10 PRINT "qqqqqqq"
    20 PRINT "ooooooo"
    30 CLEARW 2
    RUN
```

CLOSE

formato: CLOSE
CLOSE 1
CLOSE 1,2,...

Chiude un canale o tutti canali aperti in precedenza con OPEN.

CLOSEW

formato: CLOSEW n

Chiude la finestra n. Vedi anche CLEARW.

COLOR

formato: COLOR a,b,c,d,e

Definisce il colore di:
a: colore testo sulla finestra OUTPUT
b: colore sfondo finestra OUTPUT
c: colore linee
d: stile usato nel FILL
e: tipo usato nel FILL

COMMON

formato: COMMON var1,var2, ..

Dichiara le variabili che possono passare da un programma all'altro.

CONT

formato: CONT

Fa continuare un programma fermato con BREAK, STOP, CONTROL + G.

COS

formato: COS (n)

Calcola il coseno di (n). 'n' rappresenta l'angolo espresso in radianti.

```
Es. 10 PRINT COS(3.9742)
    RUN
    -.672949
```

CSNG

formato: CSNG(n)

Converte un numero di qualsiasi tipo in un reale a singola precisione.

```
Es. 10 a=4/3
    20 PRINT a
    30 b=CSNG(a)
    40 PRINT b
    RUN
    1.33333336
    1.33333
    CVI
    CVD
    CVS
```

formato: CVD (a$)
CVI (a$)
CVS (a$)

Convertono i byte di stringhe in una variabile numerica.

CVD: stringa da 8 byte
CVI: stringa da 2 byte
CVS: stringa da 4 byte

È usato per convertire i numeri ASCII da un file casuale.

D

DATA

formato: DATA n1,n2,n3
DATA a1$,a2$

Memorizza una serie di costanti che possono essere assegnate alle variabili. Può essere inserita in qualsiasi linea di programma.

```
Es. 10 DATA 3,6,8,3,1
    20 FOR i=1 TO 5
    30 READ a
    40 PRINT a
    50 NEXT
    RUN
    3
    6
    8
    3
    1
```

DEF DBL

formato: DEF DBL lettera1,lettera2

Definisce le variabili a doppia precisione.
Vedi DEF SGN

DEF FN

formato:
DEF FN nomefunzione (parametro,parametro..) = definizione

Definisce una funzione scritta dall'utente. La funzione può essere numerica o alfanumerica (in questo caso il nome della funzione termina con $). La funzione verrà chiamata con FN nomefunzione.

```
Es. 10 DEF FNsomma(x)=x+5
    20 PRINT FNsomma (2)
    30 PRINT FNsomma (7)
    RUN
    7
    13
```

È da notare che non c'è spazio tra FN e il nome della funzione.

DEFINT

formato: DEFINT lettera1-lettera2

Dichiara un insieme di lettere come variabili intere.

```
Es. 10 DEFINT c-d
    20 k=7/3
    30 c=k
    40 PRINT k
    50 PRINT c
    RUN
    2.33333
    2
```

DEFSTR

formato: DEFSTR lettera1-lettera2

Dichiara un insieme di lettere come variabili alfanumeriche.

```
Es. 10 DEFSTR b-d
    20 b="ABC"
    30 c="qwe"
    40 PRINT b,c
    RUN
    ABC qwe
```

DEFSGN

formato: DEFSGN lettera1-lettera2

Dichiara un insieme di lettere come variabili reali a singola precisione.

DELETE

formato: DELETE a-b

Cancella le linee comprese tra 'a' e 'b'.

```
Es. 10 REM
    20 REM
    30 REM
    40 REM
    50 REM
    DELETE 20-40
    LIST
    10 REM
    50 REM
```

DIM

formato: DIM nomearray,(a,b,c)
DIM nomearray$,(a,b,c)

Definisce il numero di una matrice numerica o alfanumerica.

Il numero massimo di dimensioni è 15. Il valore minimo può essere 0 o 1 modificabile con OPTION BASE. Fino a 10 elementi non è richiesto il dimensionamento.

```
Es. 10 DIM c(3)
    20 c(1)=3
    30 c(2)=7
    40 c(3)=18.47
    50 FOR i=1 TO 3
    60 PRINT c(i)
    70 NEXT i
    RUN
    3
    7
    18.47
```

```
Es. 10 DIM a$(5)
    20 a$(1)="Questa"
    30 a$(2)="è "
    40 a$(3)="una"
    50 a$(4)="frase"
    60 FOR i=1 TO 4
    70 a$(5)=a$(5)+a$(i)
    80 NEXT i
    90 PRINT a$(5)
    RUN
```

Questa è una frase

DIR

formato: DIR
DIR A:
DIR B:
DIR A:*.BAS
DIR A:*.PRG
DIR B:*.*
DIR A:?K.LOG

Elenca i file presenti su disco. 'A' o 'B' indica il drive.
'*' indica la mancanza di un dato.

Es.	
*.BAS	elenca tutti i file BASIC
*.LOG	elenca tutti i file LOGO
PROVA.*	elenca tutti i file chiamati
PROVA p*.BAS	elenca tutti i programmi BASIC che iniziano per 'P'
'?'	è come l'asterisco ma indica un solo carattere.

Es.	
?ARI.LOG	elenca tutti i file LOGO di quattro lettere che finiscono con "ARI" (es. PARI,MARI,FARI ...).

E

ED EDIT

formato: EDIT
ED
EDIT n
ED n

Chiama l'editor. Può iniziare da una linea specificata 'n'.

ELLIPSE

formato: ELLIPSE a,b,c,d,e,f

Disegna un'ellisse con i seguenti parametri.

a = coordinata x
b = coordinata y
c = raggio orizzontale
d = raggio verticale
e = angolo iniziale
f = angolo finale

I gradi sono espressi in decimi (0/3600). (vedi anche CIRCLE).

END

formato: END

Finisce un programma. Può essere un comando diretto o inserito in una linea di programma. Chiude tutti i file e ritorna alla fase di COMMAND.

EOF

formato: EOF(n)

Ritorna il valore -1 (vero) alla fine di un file sequenziale o ad accesso casuale di numero 'n'.

EOF: sono le iniziali di End Of File (fine del file).

ERA

formato: ERA nomefile
ERA A: nomefile
ERA B: nomefile

Cancella un file da disco chiamato 'nomefile'. Può essere definito anche il drive (A o B). Il file cancellato è perso per sempre quindi molta cautela nell'uso del comando.

ERASE

formato: ERASE nomematrice
ERASE nomematrice1,nomematrice2,...

Cancella una matrice precedentemente dimensionata. È neccessario usala prima di un nuovo ridimensionamento.

```
Es. 10 DIM k$(6)
    20 DIM a(5)
    30 PRINT a(4)
    40 ERASE k$,a
```

ERL

Formato: ERL

Da il numero della linea di programma in cui si è verificato un errore. Se l'errore avviene nel modo diretto ERL = 65530. La variabile ERL non può essere forzata.

```
Es. 10 ON ERROR GOTO 40
    20 PRINT 5/0
    30 END
    40 PRINT "LINEA ERRORE
       =";ERL
    RUN
    20
```

ERR

formato: ERR

Contiene il numero di codice dell'errore.

```
Es. 10 ON ERROR GOTO 40
    20 PRINT 5/0
    30 END
    40 PRINT "ERRORE=";ERR
    RUN
    11
```

ERROR

formato: ERROR n

Simula un errore di codice 'n'. 'n' deve essere un intero compreso tra 2 e 223.

```
Es. 10 ERROR 15
    RUN
    READ statement ran out of data
```

EXP

formato: EXP(n)

Ritorna l'esponente cui va elevato 'n'. Con n< =87.3365

```
Es. 10 PRINT EXP(13)
    RUN
    442413
```

FIELD

formato: FIELD n,8 AS x$ ....

Riserva lo spazio per le variabili nel buffer di un file casuale.

FILL

formato: FILL x,y,

Riempie un'area con un colore codice 'n' a partire dalle coordinate 'x' e 'y'.

```
Es. 10 COLOR 1,1,1,3,2
    20 CIRCLE 300,200,70
    30 FILL 300,200
```

FIX

formato: FIX(n)

Taglia i decimali di un numero reale.

```
Es. 10 PRINT FIX(20.37)
    RUN
    20
```

```
Es. 10 PRINT FIX(-13.9)
    RUN
    -13
```

```
Es. 10 PRINT FIX(15/7)
    RUN
    1
```

FLOAT

formato: FLOAT(n)

Trasforma un numero intero in un reale a singola precisione. Apparentemente non accade nulla ma in realtà viene riservata una quantità maggiore di memoria. 'n' dovrà essere compreso tra -32768 e +32767.

FOLLOW

formato: FOLLOW var1,var2,...

Presenta sullo schermo la linea dove si trova la variabile indicata e il suo valore.

```
Es. 10 FOR i=1 TO 5
    20 a=a+1
    30 NEXT
    FOLLOW i,a
    RUN
```

Vedi anche UNFOLLOW

FOR

formato: FOR i=n TO m
FOR i=n TO m STEP a

Crea un ciclo dal numero N al numero M eventualmente con un incremento A.
Il ciclo viene chiuso da NEXT.

n>m se step non indicato o > 1
n<m se step è negativo

```
Es. 10 FOR i=1 TO 5
    20 PRINT i
    30 NEXT i
    RUN
    1
    2
    3
    4
    5
```

```
Es. 10 FOR i=1 TO 5 STEP .2
    20 PRINT i
    30 NEXT i
    RUN
    1
    1.2
    1.4
    1.6
    .
    .
    .
```

FRE

formato: FRE(0)

Ritorna la quantità di memoria (in byte) ancora libera.

```
Es. PRINT FRE(0)
```

FULLW

formato: FULLW n

Dimensiona la finestra indicata per la massima dimensione. Per il significato di 'n' vedi CLEARW.

GET

formato: GET numero file,numero record

Legge un record da un file casuale da disco.

GOSUB

formato: GOSUB n
GOSUB nomelabel

Salta ad una subroutine indicata dal numero di linea 'n' o dal nome della label. La LABEL deve essere definita con un nome seguito da ':' .

```
Es. 10 GOSUB 300
    20 PRINT "a"
    30 GOSUB 300
    40 END
    300 PRINT "b"
    310 RETURN
    RUN
    b
    a
    b
```

```
Es. 10 PRINT "calcolo somma"
    20 a=30
    30 b=15
    40 GOSUB calcolo
    50 PRINT c
    60 END
    100 calcolo:
    110 c=a+b
    120 RETURN
    RUN
    45
```

GOTO

formato: GOTO n
GOTO nomelabel

Salto incondizionato ad una determinata linea 'n' o al nome della LABEL.

```
Es. 10 PRINT "linea 10"
    20 GOTO 50
    30 END
    50 PRINT "linea 50"
    RUN
    linea 10
    linea 50
```

GOTOXY

formato: GOTOXY x,y

Forza la posizione di stampa alle coordinate x,y.

```
Es. 10 GOTOXY 4,5
    20 PRINT"o"
```

H

HEX$

formato: HEX$(n)

Ritorna una rappresentazione esadecimale del numero decimale N che dovrà essere compreso tra -32768 e +32767.

I

INP

formato: INP(n)

Ritorna il valore letto alla portà N. Il numero 'n' ha il seguente significato:

0: STAMPANTE
1: RS 232
2: CONSOLE
3: MIDI
4: TASTIERA

INPUT

formato: INPUT n
INPUT a$
INPUT "testo";a
INPUT "testo";a$
INPUT n,a
INPUT n,a$

Acquisisce un dato da tastiera o da un canale indicato dopo il simbolo del cancelletto.

```
Es. 10 PRINT "Dammi un numero"
    20 INPUT n
    30 PRINT n
```

```
Es. 10 INPUT "Dammi il numero";n
    20 PRINT n
```

```
Es. 10 INPUT "Dammi il nome";a$
    20 PRINT "Nome = ";a$
```

INPUT$

formato: INPUT$(n)
INPUT$(n),a

Ritorna un numero N di caratteri letti da tastiera o da un file di dati.

INSTR

formato: INSTR(n,a$,b$)

Ritorna la posizione in cui si trova la stringa B$, nella stringa A$ a partire dalla posizione N. Se la combinazione di caratteri non viene trovata il risultato è uguale a zero. N è un intero compreso tra 1 e 255. Se N non è indicato si inizia dal primo carattere.

```
Es. 10 a$ = "Testo principale"
    20 b$ = "inc"
    30 t = INSTR(1,a$,b$)
    40 PRINT t
    RUN
    9
```

INT

formato: INT(n)

Trasforma il numero reale N in un intero.

```
Es. 10 PRINT INT(53.6)
    RUN
    53
```

KILL

formato: KILL "nomefile"

Cancella un file da disco.

LEFT$

formato: LEFT$ (a$,n)

Ritorna l'espresssione alfanumerica costituita da N caratteri più a sinistra di a$. Se n=0 la stringa risultante è nulla. N dovrà essere compresa tra 1 e 255.

```
Es. 10 a$ = "abcdefg"
    20 t$ = LEFT$(a$,4)
    30 PRINT t$
    RUN
    abcd
```

LEN

formato: LEN(a$)

Ritorna il numero N di caratteri della stringa a$.

```
Es. 10 PRINT LEN("abc")
    RUN
    3
```

LET

formato: LET n = x
LET n$ = "stringa"

Assegna il valore ad una variabile. Il valore deve essere compatibile con il tipo di variabile. È un'istruzione opzionale può cioè essere omessa.

LET a = 5 corrisponde: a = 5

```
Es. 10 LET a=4
    20 PRINT a
    RUN
    4
```

LINEF

formato: LINEF(x1,y1,x2,y2)

Disegna una linea tra i punti (x1,y1) e (x2,y2). La posizione 0,0 si trova in alto a sinistra. La dimensione dello schermo dipende dalla risoluzione e dal numero di colori voluti.

```
Es. 10 LINEF 40,20,300,200
    RUN
```

LINE INPUT

formato: LINE INPUT variabile
LINE INPUT n,variabile

Legge da un file da disco sequenziale e assegna alla variabile stringa il valore.

LIST

formato: LIST
LIST n
LIST n,m,v
LIST n-m
LIST n-m,k-v

Esegue la lista di un programma sulla finestra LIST.

LIST	visualizza tutto il programma
LIST n	visualizza la sola linea N
LIST n,m,v	visualizza le linee indicate
LIST n-m	visualizza le linee comprese tra N e M
LIST n-m,k-v	visualizza le linee comprese

LLIST

formato: LLIST
LLIST n
LLIST n,m,v
LLIST n-m
LLIST n-m,k-v

Esegue la lista di un programma su carta.

LLIST	stampa tutto il programma
LLIST n	stampa la sola linea N
LLIST n,m,v	stampa le linee indicate
LLIST n-m	stampa le linee comprese tra N e M
LLIST n-m,k-v	stampa le linee comprese

Il numero di colonne inizialmente è uguale a 72 ma può essere modificato con:

WIDTH LPRINT n

dove N è il numero di colonne.

LOAD

formato: LOAD"nomeprogramma"

Carica nella memoria un programma precedentemente memorizzato su disco.

LOC

formato: LOC(n)

Ritorna il numero di record e/o il numero di byte letti o scritti in un file di numero N.

LOF

formato: LOF(n)

Ritorna il numero di byte di cui è composto un file di numero N.

LOG

formato: LOG(n)

Restituisce il logaritmo naturale del numero N che in ogni caso dovrà essere maggiore di 0.

```
Es. 10 PRINT LOG(103)
    RUN
    4.63472
```

LOG10

formato: LOG10(n)

Restituisce il logaritmo a base 10 del numero N anch'esso maggiore di 0.

```
Es. 10 PRINT LOG10(103)
    RUN
    2.01283
```

LPOS

formato: LPOS(0)

Ritorna la posizione della testina della stampante dopo l'ultimo ritorno del carrello.

LPRINT

formato: LPRINT
LPRINT a$
LPRINT USING "formato";n

Manda alla stampante la stringa a$ oppure usando il formato definito da USING. Il numero massimo di colonne può essere definito da:

WIDTH LPRINT n

dove N è il numero di colonne.

LSET

formato: LSET var1$ = var2$

Muove una stringa all'interno di una specificata variabile stringa senza riassegnarla.

M

MERGE

formato: MERGE "nomefile"

Fonde nella memoria del calcolatore il programma esistente con un'altro caricato da disco. Il programma deve essere scritto in BASIC.

MID$

formato: MID$(a$,a,b)

Ritorna l'espressione alfanumerica composta di B caratteri a partire dalla posizione A della stringa a$. Se A è maggiore della lunghezza di a$ la stringa ritorna vuota.

```
Es. 10 a$="abcdefgh"
    20 t$=MID$(a$,3,4)
    30 PRINT t$
    RUN
    cdef
```

formato: a$=MKI$(a)
a$=MKS$(a)
a$=MKD$(a)

MKI$: converte un intero in una stringa a 2 byte
MKS$: converte un reale a singola precisione in una stringa a 4 byte
MKD$: converte un reale a doppia precisione in una stringa a 8 byte.

NAME

formato: NAME "nomefilevecchio" AS "nomefilenuovo"

Impone un nuovo nome ad un file esistente.

NEXT

formato: NEXT
NEXT a
NEXT a,b,...

Chiude un ciclo aperto con FOR ... TO. Se non è indicata la variabile, viene chiuso l'ultimo ciclo. Si possono indicare anche più variabili di ciclo separate da una virgola.

```
Es. 10 FOR i=1 TO 5
    20 FOR j=1 TO 2
    30 PRINT i+j
    40 NEXT j,i
    RUN
    2
    3
    3
    4
    4
    5
    .
    .
```

NEW

formato: NEW
NEW nomeprogramma

Cancella un programma dalla memoria. Questo comando deve essere usato con cautela perchè la perdita del programma è irrimediabile.

OCT$

formato: OCT$(n)

Restituisce una rappresentazione ottale del numero decimale N che deve essere un intero compreso tra -32768 e +32767.

```
Es. 10 a$=OCT$(13)
    20 PRINT a$
    RUN
    15
```

OLD

formato: OLD "nomefile"

Carica un programma da disco. Ha lo stesso significato di LOAD.

ON

formato: ON n GOTO a,b,c
ON n GOSUB a,b,c,...
ON ERROR GOTO a

Il programma salta alla linea indicata dopo GOTO o GOSUB in base al numero che è presente nella variabile N.

```
Es.  10 INPUT "numero tra 1 e 3";
        on
     20 ON n GOTO 100,300,200
     30 END
    100 PRINT "numero 1":END
    200 PRINT "numero 3":END
    300 PRINT "numero 2":END
```

L'istruzione ON ERROR GOTON fa saltare il programma alla linea N non appena ha trovato una situazione di errore. Il programma esempio che segue salta alla linea 100 in quanto è stato commesso un errore di sintassi alla linea 30 (PRUNT invece di PRINT).

```
Es.  10 ON ERROR GOTO 100
     20 PRINT "a"
     30 PRUNT "o"
     40 END
    100 PRINT "errore"
```

OPEN

formato: OPEN "O",n,"nomefile",t
OPEN "I",n,"nomefile",t
OPEN "R",n,"nomefile",t

Apre un canale di numero N e con il nome "nomefile", T rappresenta la lunghezza del record. Il valore è opzionale. Il modo è definito come segue.

O: OUTPUT per un file sequenziale

I: INPUT per un file sequenziale

R: INPUT/OUTPUT per un file casuale (RANDOM)

OPENW

formato: OPENW n

Apre una finestra di numero N secondo la convenzione vista in CLEARW.

OPTION BASE

formato: OPTION BASE 0
OPTION BASE 1

Determina il valore minimo dell'indice di una matrice.

OUT

formato: OUT a,b

Trasmette il valore B ad una porta A . Il valore N dovrà essere compreso tra 0 e 255. Il valore A sarà dedotto dalla seguente tabella:

0: porta parallela
1: RS232
2: console
4: MIDI
4: tastiera

PCIRCLE

formato: PCIRCLE x,y,r,a,b

Come per l'istruzione CIRCLE ma questa volta la figura è riempita con l'ultimo colore selezionato. Per le variabili fare riferimento all'istruzione CIRCLE.

PELLIPSE

formato: PELLIPSE a,b,c,d,e,f

Come per l'istruzione ELLIPSE ma questa volta la figura è riempita con l'ultimo colore selezionato. Per le variabili fare riferimento all'istruzione ELLIPSE.

PEEK

formato: PEEK (n)

Legge il contenuto di una locazione di memoria N. Il tipo di valore ricavato dipende dall'ultima definizione di DEFSEG.

DEFSGN	PEEK
>0	1 Byte
=0	2 Byte
<0	4 Byte

```
Es. 10 PRINT PEEK(16544)
    RUN
    16385
```

POKE

```
Es. POKE n,m
```

Scrive un numero N in una locazione di memoria M. Il tipo di numero N dipende dalla tabella seguente:

DEFSGN	POKE
>0	1 Byte
=0	2 Byte
<0	4 Byte

ATTENZIONE. Un uso non corretto dell'istruzione può portare al blocco del sistema. In tal caso spegnere il computer e riaccendere. Questa operazione provoca la perdita di eventuali programmi in memoria.

POS

POS formato: POS(0)

Ritorna la posizione attuale del cursore sullo schermo o sulla stampante.

```
Es. 10 PRINT POS(0)
    20 PRINT " ";
    30 PRINT POS(0)
    RUN
```

PRINT

formato: PRINT
PRINT 3+5
PRINT "qwert"
PRINT a,a$
PRINT n,a,a$...

Stampa sullo schermo quello che appare dopo il comando.
Possono essere stampate variabili numeriche e alfanumeriche. Il simbolo del cancelletto seguito da un numero di canale permette di inviare il contenuto ad una periferica diversa dallo schermo. Tale periferica deve essere prima assegnata con l'istruzione OPEN.

```
Es. 10 PRINT "aaaaa"
    20 PRINT 3+8
    40 PRINT
    60 PRINT 3
```

Il comando può essere sostituito dal punto interrogativo.

```
Es. 10 ? "www"
```

È possibile stabilire anche il formato di stampa con l'istruzione:

PRINT USING ".";n

```
Es. 10 a=12.1234
    20 PRINT USING ".";a
    30 PRINT USING ".";a
    RUN
    12.1
    12.12
```

Viene stabilita la forma da dare ai decimali.
Si possono introdurre tra apici anche altri simboli:

```
USING "& &"
```

Indica che il numero di caratteri stampati sarà pari al numero di spazi compresi tra le 'e' commerciali più due.

```
Es. 10 PRINT USING "& &";
       "abcdefghi"
    RUN
    abcde
```

```
USING "!"
```

Stampa solo il primo carattere della stringa.

```
Es. 10 PRINT USING "!";"abc"
    RUN
    a
USING "??"
```

Vengono sostituiti i simboli ?? con le espressioni date.

```
Es. 10 PRINT USING "?? 520 ??";
       "ATARI","ST"
    RUN
    ATARI 520 ST
```

```
USING "!"
```

Indica dove deve essere stampato il punto decimale.

```
USING ","
```

Quando viene inserita la virgola alla sinistra del punto decimale verrà stampata la virgola davanti ad ogni gruppo di 3 cifre alla sinistra del punto decimale.

```
USING "+"
```

Indica che verrà stampato il simbolo + o - davanti al numero.

```
USING "-"
```

Questo simbolo può essere usato solo al termine del formato di stampa. Il risultato sarà la stampa di uno spazio dopo un'espresssione positiva ed un segno meno dopo un'espressione negativa.

```
USING "**"
```

Devono essere inseriti all'inizio del formato di stampa facendo si che gli spazi di testa siano riempiti con asterischi anzichè con zeri.

USING "$$"

I segni dollaro possono essere usati solo all'inizio del formato di stampa. Viene stampato un dollaro davanti all'espresssione.

USING "^^^^"

L'espressione viene stampata in formato esponenziale. Il punto decimale può essere stampato ovunque.

PUT

formato: PUT n,numerorecord

Scrive un record su un file casuale su disco.

Q

QUIT

formato: QUIT

Vedi SYSTEM

R

RANDOMIZE

formato: RANDOMIZE n

Altera la variabile di controllo usata per generare il valore di RND. N è una funzione con risultato compreso tra 0 e 65535.

READ

formato: READ n,a$

Legge i valori numerici, alfanumerici memorizzati in DATA.

REM

formato: REM commento
' commento

Permette di inserire i commenti all'interno del programma.
La linea non ha alcuna influenza sul programma.

RENUM

formato: RENUM a,b,c
RENUM

Rinumera le linee di un programma a partire dalla linea A, B è la riga del programma esistente da cui deve iniziare la numerazione. C è il passo di cui va incrementato il numero di linea.

```
Es. 3 REM
    5 REM
    7 PRINT
    RENUM
    LIST
    10 REM
    20 REM
    30 PRINT
```

Se non indicati i parametri, la numerazione inizia da 10 e si incrementa di 10 in 10. Durante l'operazione deve essere inserito un disco formattato e non protetto.

REPLACE

formato: REPLACE "nomefile",elenconumerilinea

Sostituisce una vecchia versione di un file con una nuova .

RESET

formato: RESET

Permette di cambiare disco quando il programma è in funzione.

RESTORE

formato: RESTORE
RESTORE n

Legge le costanti da un' istruzione DATA a partire da una linea di programma n. Quando non è indicata si inizia dalla prima linea contenete DATA.

RESUME

formato: RESUME (0)
RESUME numerolinea
RESUME NEXT

Continua l'esecuzione dopo un errore. Se è indicato il numero di linea, dalla linea stessa. Se è indicato NEXT dalla riga dopo l'errore. Con (0) dalla stessa linea dove è avvenuto l'errore.

RETURN

formato: RETURN

Chiude una subroutine chiamata con GOSUB. La linea eseguita sarà quella successiva al GOSUB. La subroutine può essere chiamata da GOSUB n oppure da ON n GOSUB.

RIGHT$

formato: RIGHT$ (a$,n)

Ritorna l'espressione alfanumerica costituita da N caratteri più a destra di a$. Se n=0 la stringa risultante è nulla. N dovrà essere compresa tra 1 e 255.

```
Es. 10 a$="abcdef"
    20 t$=RIGHT$(a$,3)
    30 PRINT t$
    RUN
    def
```

RND

formato: RND
RND (0)
RND (n)
RND (-n)

RND restituisce il numero casuale compreso tra 0 e 1.
Scrivendo RANDOMIZE prima dell'istruzione RND la generazione casuale sarà uguale ad ogni lancio di programma.

RND (0): Ritorna l'ultimo numero generato.
RND (n): Ritorna il primo numero nella sequenza attuale.
RND (-n): Ritorna il primo numero di una nuova sequenza.

```
Es. 10 PRINT RND (1)
    RUN
    .850978
```

RSET

formato: RSET variabile stringa = espressione stringa

Muove una stringa all'interno di una variabile stringa specificata senza riassegnarla.

RUN

formato: RUN
RUN,n
RUN nomeprogramma.BAS
RUN nomeprogramma.BAS,n

Manda in esecuzione un programma presente in memoria.

RUN n: il numero N è la linea da cui deve avere inizio il programma
RUN nom.BAS: Manda in esecuzione un programma (denominato nom-.BAS) presente su disco

Il programma può essere interrotto selezionando BREAK nel menù discendente RUN. Per far continuare un programma scrivere CONT.

SAVE

formato: SAVE "nomefile"
SAVE "nomefile",numeri linea

Salva un programma su disco. Il disco deve essere formattato e non protetto. Indicando il numero delle linee, solo queste vengono salvate.

SGN

formato: SGN (n)

Ritorna il segno del numero N.

```
Es. 10 PRINT SGN (6)
    RUN
    1
```

```
Es. 10 PRINT SGN (-7)
    RUN
    -1
```

SIN

formato: SIN (a)

Ritorna il seno dell'angolo A.

SOUND

formato: SOUND a,b,c,d,e

Genera un suono con questi parametri:

a: voce tra 1 e 3, 4 indica un rumore
b: volume
c: nota
d: ottava
e: durata

```
Es. 10 FOR i=1 TO 15
    20 FOR j=1 TO 15
    30 SOUND 1,8,i,j,9
    40 NEXT j,i
```

SPACE$

formato: SPACE$ (n)

Ritorna una stringa formata da N spazi con N compreso tra 0 e 255.

```
Es. 10 t$=SPACE$(6)
    20 PRINT "a";t$;"b"
    RUN
    a      b
```

SPC

formato: PRINT SPC (n)
PRINT .. SPC (n)
LPRINT SPC (n)

Stampa un numero N di spazi. Se N è maggiore del numero di colonne (nc) verranno stampate (nc-n) colonne.

SQR

formato: SQR (n)

Calcola la radice quadrata del numero N che dovrà essere maggiore o uguale a 0.

```
Es. 10 PRINT SQR (13)
    RUN
    3.60555
```

STRING$

formato: STRING$ (a,b)
STRING$ (a,a$)

Ritorna una stringa formata da A caratteri corrispondente al codice ASCII B o al primo carattere della stringa a$. Il numero A dovrà essere compreso tra 0 e 255.

```
Es. 10 PRINT STRING$(6,65)
    RUN
    AAAAAA
```

```
Es. 10 PRINT STRING$ (6,"prova")
    RUN
    pppppp
```

STR$

formato: STR$ (n)

Ritorna una rappresentazione alfanumerica dell'espressione numerica N.

```
Es. 10 k=3.14*9+2
    20 v$=STR$ (k)
    30 PRINT v$
    RUN
    30.2599
```

STEP

formato: FOR i=x TO y STEP n

Determina il passo di cui deve essere incrementato il ciclo FOR .. NEXT. N è un numero reale.

```
Es. 10 FOR i=1 TO 20 STEP .2
    20 PRINT i
    30 NEXT
    RUN
```

```
Es. 10 FOR i=20 TO 1 STEP -1
    20 PRINT i
    30 NEXT
    RUN
```

STEP

formato: STEP
STEP,n
STEP nomeprogramma
STEP nomeprogramma,n

Esegue un programma linea per linea premendo il tasto RETURN. N è il numero della prima linea da eseguire.

STOP

formato: STOP

Ferma un programma in esecuzione e restituisce il controllo alla finestra COMMAND. CONT o <RETURN> fa ricontinuare il programma.

SWAP

formato: SWAP a,b
SWAP a$,b$

Scambia i valori di due variabili tra loro.

```
Es. 10 a=5
    20 b=3
    30 SWAP a,b
    40 PRINT a,b
```

SYSTAB

formato: PEEK (SYSTAB + n)
POKE (SYSTAB + n,x)

La variabile SYSTAB indica l'inizio delle locazioni di memoria della tabella di sistema. N è un numero compreso tra 0 e 24.

N.	mod.	lun.	funzione		
0	N	2	Risoluzione grafica ←		1 ALTA 2 MEDIA 3 BASSA
2	S	2	EDITOR GHOST LINE STYLE		
4	N	2	EDIT AES		
6	N	2	LIST		
8	N	2	OUTPUT		
10	N	2	COMMAND		
12	N	2	EDIT OPEN FLAG ←		0 CHIUSO 1 APERTO
14	N	2	LIST OPEN FLAG ←		0 CHIUSO 1 APERTO
16	N	2	OUTPUT OPEN FLAG ←		0 CHIUSO 1 APERTO
18	N	2	COMMAND OPEN FLAG ←		0 CHIUSO 1 APERTO
20	N	2	GRAPHIC BUFFER		
24	N	4	GEM FLAG ←	0=ON 1=OFF	Quando si trova in OFF il video, il mouse e la tastiera sono disinseriti ma la velocità di elaborazione è maggiore.

N : È il numero da sommare.
mod : Indica se la variabile è modificabile o meno (S=si, N=no).
lun : è la lunghezza in byte.

SYSTEM

formato: SYSTEM

Ritorna il controllo al GEM eliminando il BASIC e perdendo ogni programma in memoria. Questo comando va quindi usato con cautela. Vedi anche QUIT.

T

TAB

formato: PRINT TAB (n)
PRINT ... TAB (n)
LPRINT TAB (n)

Sposta il cursore ad una posizione specificata. Il numero N dovrà essere compreso tra -32768 e +32767.

```
Es. 10 PRINT TAB (15);"a"
```

TAN

formato: TAN (n)

Restituisce la tangente del numero N.

THEN

formato: IF condizione THEN istruzione

Determina la condizione descritta da IF. L'istruzione può essere sostituita dal numero di linea a cui saltare.

```
Es. 10 INPUT "Dammi un numero
       tra 1 e 5 ";a
    20 IF a=5 THEN 50
    30 ? "RIPETI"
    40 GOTO 10
    50 ? "OK"
```

Il programma continua a chiedere il numero fino a quando non viene introdotto il 5.

TRACE

formato: TRACE
TRACE i,j
TRACE i-j

Esegue un programma passo passo per la messa a punto.
Vengono stampate tutte le linee eseguite.

TRACE

Formato: TRACE : esegue tutte le linee
TRACE i,j : esegue solo le linee indicate
TRACE i-j : esegue le linee comprese

```
Es. 10 REM
    20 REM
    30 PRINT
    40 REM
    TRACE
    RUN
```

TROFF

formato: TROFF
TROFF numeri linea

Cancella il comando TRON. Si possono selezionare le linee. Vedi anche TRON.

TRON

formato: TRON
TRON n,j
TRON n-j

Usato durante la fase di messa a punto dei programmi stampa tra parentesi quadre il numero di linea eseguito. N e J sono i numeri di linea da cui inizia la TRON (TRace ON).

```
Es. 10 REM
    20 REM
    30 REM
    40 REM
    50 REM
    TRON
    RUN
```

Vedi anche TROFF, TRACE, UNTRACE, FOLLOW, UNFOLLOW

U

UNBREAK

formato: UNBREAK
UNBREAK numeri di linea

Cancella il comando BREAK. Le variabili n,k,j indicano le linee selezionate.

UNFOLLOW

formato: UNFOLLOW

Termina la presentazione delle variabili. Chiude la stampa aperta da FOLLOW.

UNTRACE

formato: UNTRACE
UNTRACE numerilinea

Chiude la funzione TRACE. È possibile selezionare le linee di programma.

VAL

formato: VAL (a$)

Trasforma una stringa alfanumerica in numerica se formata da numeri.

```
Es. 10 a=VAL ("5.657")
    20 PRINT a
    RUN
    5.657
```

VARPTR

formato: VARPTR (n)
VARPTR (k)

Ritorna l'indirizzo di una variabile o il numero assegnato ad un file su disco quando viene aperto. Il valore ricavato è l'indirizzo del primo byte del nome della variabile.

```
Es. 10 a=5
    20 PRINT VARPTR (a)
    RUN
    447812
```

W

WAIT

formato: WAIT n,a,b

Ferma il programma fino a quando non c'è un segnale in una porta (di numero N).

WAWE

formato: WAWE a,b,c,d,e

Controlla la forma d'onda usata dall'istruzione SOUND.

a: attiva il registro mixer.

bit 0 = voce 1
bit 1 = voce 2
bit 2 = voce 3
bit 3 = attiva il rumore nel canale 1
bit 4 = attiva il rumore nel canale 2
bit 5 = attiva il rumore nel canale 3
L'attivazione avviene portando a 0 il bit relativo.

b: inviluppo. È ottenuto portando ad uno il bit relativo secondo questo schema.
bit 0: canale 1
bit 1: canale 2
bit 2: canale 3

c: Determina la forma d'onda. Sono usati i bit da 0 a 3.

d: Periodo. Determina il periodo d'inviluppo.

e: Ritardo in cinquantesimi di secondo prima di restituire il controllo al BASIC.

WEND

formato: WEND

Segnala la fine di un ciclo WHILE/WEND.

```
Es. 10 WHILE a
    20 INPUT "Dammi un
       numero ";a
    30 WEND
    40 END
```

Il programma termina scrivendo il numero 0.

WHILE

formato: WHILE espressionelogica

Condizione di controllo di un ciclo WHILE/WEND. Ripete quello che è compreso fino a quando il valore della variabile di controllo è uguale a 0.

```
Es. 10 e=5
    20 WHILE e
    30 e=e-1
    40 PRINT e
    50 WEND
    60 END
    RUN
```

WIDTH

formato: WIDTH n
WIDTH LPRINT n

Stabilisce il numero di colonne che si usano sul video o sulla stampante. N dovrà essere compreso tra 14 e 255.

WRITE

formato: WRITE a,b,a$
WRITE n,a,b,a$

Come per PRINT. Se si usa un'uscita diversa dal video deve essere aperto un canale con modalità 'O' o 'R'. Vedi anche il comando PRINT.

Messaggi di errore

Durante l'uso dei programmi scritti in BASIC si possono verificare delle situazioni di errore segnalate da altrettanti messaggi. Diamo di seguito la traduzione italiana con accanto il relativo numero di codice. Il numero può essere richiesto in maniera diretta con PRINT ERR.

COD.	DESCRIZIONE
0	Nessun errore
2	C'è un errore
3	Il comando RETURN abbisogna di GOSUB
4	Il comando READ manca di dati
5	Chiamata una funzione non permessa
6	Numero troppo grande
7	Il programma è troppo grande per la memoria
8	Il comando è riferito ad una linea di programma non esistente
9	L'indice supera quello della matrice
10	Hai definito una matrice più di una volta
11	Non puoi dividere per zero
12	Non si può usare il comando in modo diretto
13	Tipi di valori non compatibili
15	La stringa non può avere più di 255 caratteri
16	Espressione troppo lunga o troppo complessa
17	CONT può essere usato solo in modo BREAK
20	RESUME è stato trovato prima di essere entrati in una routine di errore
22	L'espressione ha l'operatore che non segue l'operando
23	Linea di programma troppo grande
50	Il comando FIELD ha causato un overflow
52	Il numero o il nome del file non è corretto
53	File non trovato nel drive specificato
54	Il modo file non è valido
55	Non puoi usare OPEN o KILL con un file aperto
57	Errore nella fase input/output da disco. Ripetere l'operazione
58	File non esistente
61	Il disco è pieno
62	Hai superato la fine del file (End Of File)
63	Il numero del record usato con PUT o GET è maggiore di 32767 o è 0
64	Nome del file non valido
65	Carattere non valido in file programma
66	Il file programma ha comandi senza numeri di linea
99	BREAK
101	Il programma eccede la dimensione della memoria

segue

seguito

102	Il comando ON è fuori range
104	È richiesta una variabile
106	Non esiste il numero di linea
107	Il numero è troppo grande per un integer
108	Il dato di input non è valido. Riinserisci il numero dall'inizio
109	STOP
110	Hai delle subroutine annidate troppe volte
111	File BLOAD non valido
202	Comando non permesso in questo punto
203	Richiesto il numero di linea
204	Il comando FOR abbisogna di NEXT o il comando WHILE abbisogna di WEND
205	Il comando NEXT abbisogna di FOR o il comando WEND abbisogna di WHILE
206	Manca una virgola
207	Manca una parentesi
208	Il comando OPTION BASE Può avere solo 1 o 0
209	Manca il comando di fine
210	Argomento troppo grande in list
212	Non puoi ridefinire la variabile
213	Funzione definita più di una volta
214	Hai provato a saltare dentro un ciclo
221	Errore di sistema X , prego ripartire
222	Programma non mandato in esecuzione
223	Troppi cicli FOR

Esempi di programmi Basic

La raccolta di programmi che segue vuole essere uno stimolo per il lettore affinchè possa imparare tutti i segreti della programmazione in linguaggio BASIC. I programmi affrontano i temi più vari e devono essere visti come un punto di partenza per uno sviluppo più personale e complesso. Capire il funzionamento dei programmi significa impadronirsi delle conoscenze spesso difficili se non impossibili da trasmettere a parole. In alcune versioni del BASIC sarà neccessario eliminare il BUF GRAPHICS (come descritto in precedenza) in modo da permettere al programma di girare comodamente. Dopo essere entrati in EDIT si può iniziare la ricopiatura dei programmi e quindi il loro salvataggio su disco. Questa operazione è bene sia effettuata sempre prima di far partire il programma in quanto alcuni errori potrebbero mandare in blocco il sistema e far perdere il programma e con esso alcune ore di lavoro. La ricopiatura riga per riga è senza dubbio il modo migliore per imparare a conoscere i programmi. È un'operazione simile alla trascrizione di testi in lingua straniera per impararne il significato. Assieme ad ogni programma sono riportate informazioni sulla sua lunghezza, l'elenco delle variabile, commenti alle linee e consigli per un loro ulteriore sviluppo.
Una volta mandati in esecuzione i programmi possono essere bloccati premendo contemporaneamente i tasti 'CONTROL' e 'G'.
Data la diversa mappa di memoria, in alcuni listati spesso il simbolo "'" è sostituito da ":" e il simbolo "#" dal simbolo "?".

Programma n. 1

TITOLO : CALCOLO PIANI DI AMMORTAMENTO
LUNGHEZZA : 2983 Byte

DESCRIZIONE:

Il programma esegue il calcolo e la stampa di un piano di ammortamento di un capitale fino a 999.999.999. Il piano di ammortamento è l'estinzione graduale di un debito con una serie di quote costanti annuali. L'uso del programma è particolarmente facile. Si tratta di inserire in sequenza il valore del debito, il numero di anni in cui si intende estinguere il debito stesso e la percentuale di interesse. Il programma calcolerà la quota capitale, la quota interesse, il debito estinto e il debito residuo anno per anno. Vengono stampati fino a 7 anni in uno stesso schermo. Se il periodo di ammortamento è maggiore è sufficiente premere il tasto <RETURN> per continuare con la stampa.

Ecco un esempio:

Valore del debito : 5.200.000
Anni di estinzione debito : 4
Percentuale di interesse : 12.5%

Anno	Rata annua	Quota Capitale	Quota Interessi	Debito Estinto	Debito Residuo
1	1730082	1080082	650000	1080082	4119918
2	1730082	1215092	514990	2295174	2904826
3	1730082	1366977	363103	3662151	1537849
4	1730082	1537851	192231	5200001	2

Il risultato finale del debito residuo non è uguale a 0 per un errore introdotto negli arrotondamenti. Si tratta comunque di un errore irrisorio rispetto all'ammontare totale del capitale. Il programma è previsto per il solo uso del monitor ma con poche modifiche è in grado di copiare gli schermi sulla stampante.
Il programa utilizza le seguenti formule:

$$\text{Quota amm.} = \frac{C*r*q\hat{\ }n}{q\hat{\ }n-1}$$

C: capitale
r: saggio interesse
q: (r + 1)
n: numero di anni

COMMENTO SINGOLE ISTRUZIONI	
10/170	Inizializzazione delle variabili e pulizia dello schermo.
180/360	Input dei dati e verifica della correttezza degli stessi.
370/820	Calcolo dei dati e stampa sullo schermo.
830/970	Stampa intestazione.
980/1100	Routine nuova pagina.

ELENCO VARIABILI	
sp	numero di spazi bianchi
i	ciclo caricamento stringa/stampa
t$	stringa per la grafica
c	valore del debito
n	numero anni da estinguere debito
r	percentuale di interesse
q	r + 1
e	numero di anni
a	debito estinto
b	valore del capitale
s	quota annua costante
f	quota capitale
g	quota interessi
d	debito residuo
k	variabile di lavoro
u	ciclo stampa righe mancanti
i$	lettura tastiera

```
10   rem   ******************************
20   rem   *                            *
30   rem   *       PROGRAMMA n. 1       *
40   rem   *                            *
50   rem   *       CALCOLO PIANO        *
60   rem   *     DI AMMORTAMENTO        *
70   rem   *                            *
```

```
80     rem     ********************************
90     rem
100    sp=20
110    fullw 2
120    clearw 2
130    for i%=1 to 71
140    t$=t$+"-"
150    next
160    w$(1)="|----|------------|------------|--
------------|------------ -"
170    w$(2)=w$(1)+"|------------|"
180    clearw 2
190    gotoxy 0,5
200    ? spc (sp);"CALCOLO PIANO DI AMMORTAMENTO
"
210    ?:?:?
220    ?
230    ? spc(sp);
240    ? "Dammi il valore del debito : ";
250    input c
260    if c<=0 or c>999999999 then 220
270    ?
280    ? spc(sp);
290    ? "In quanti anni vuoi estinguere il debi
to ";
```

```
300   input n
310   if n<1 or n>50 then 270
320   ?
330   ? spc(sp);
340   ? "Dammi la percentuale di interesse ";
350   input r
360   if r<1 or r>100 then 320
370   rem
380   rem     ******************************
390   rem     *                            *
400   rem     *    CALCOLO E STAMPA DATI   *
410   rem     *                            *
420   rem     ******************************
430   rem
440   r=r/100
450   q=1+r
460   e=n
470   a=0
480   b=c
490   s=c*r*q^n/(q^n-1)
500   gosub 830
510   for i=1 to n
520   e=n-i+1
```

```
530   f=c*r/(q^e-1)
540   g=c*r
550   a=a+f
560   d=b-a
570   ? w$(2);
580   ? "¦ ";
590   ? using "##";i;
600   ? " ¦ ";
610   ? using "#########";s;
620   ? "  ¦ ";
630   ? using "#########";f;
640   ? "  ¦ ";
650   ? using "#########";g;
660   ? "  ¦ ";
670   ? using "#########";a;
680   ? "  ¦ ";
690   ? using "#########";abs(d);
700   ? "  ¦";
710   c=d
720   if int(i/7)=(i/7) and n<>i then gosub 980
730   next
740   ? t$
750   if int(n/7)=(n/7) then ? : goto 800
```

```
760   k=(7-((n/7)-int(n/7))*7)*2+1

770   for u=1 to k

780   ?

790   next

800   ? spc(sp);"Premi <RETURN> per continuare
... ";

810   input i$

820   goto 180

830   rem

840   rem     ******************************

850   rem     *                            *

860   rem     *     STAMPA INTESTAZIONE    *

870   rem     *                            *

880   rem     ******************************

890   rem

900   clearw 2

910   gotoxy 0,0

920   ? t$;

930   ? "|Anno| Rata annua |   Quota    |   Quo
ta    |   Debito   ";

940   ? "|   Debito   |";

950   ? "|    |            | Capitale   |  Inte
ressi |   Estinto  ";

960   ? "|   Residuo  |";

970   return
```

```
980   rem
990   rem    *******************************
1000  rem   *                             *
1010  rem   *        NUOVA PAGINA         *
1020  rem   *                             *
1030  rem   *******************************
1040  rem
1050  ? t$
1060  ?
1070  ? spc(sp);"Premi <RETURN> per continuare
... ";
1080  input "";i$
1090  gosub 830
1100  return
```

Programma n. 2

TITOLO : DIAGRAMMA A BARRE
LUNGHEZZA : 8557 Byte

DESCRIZIONE:

La rappresentazione grafica di una serie di valori è uno dei mezzi più semplici e utilizzati per esprimere la tendenza di un fenomeno. Il caso più tipico sono le vendite di un prodotto nel corso dei vari periodi dell'anno. L'andamento rappresentato mediante un grafico è utilissimo per trasmettere ad altri questi valori. Il programma è in grado di caricare una serie di dati e salvarli su disco per successive elaborazioni. Vediamo in dettaglio l'utilizzo del programma.

Dopo aver caricato il programma da disco e averlo mandato in esecuzione con <RUN>, verrà presentato un menù con 3 opzioni a disposizione:

1. INPUT DATI DA TASTIERA

Permette il caricamento dei valori direttamente dalla tastiera. Viene chiesto il nome dei dati (non oltre otto caratteri). Vengono chiesti in sequenza i valori di ogni mese. Alla fine del caricamento è sufficiente premere <RETURN> per passare al secondo menù. I valori dovranno in ogni caso essere positivi al fine di un corretto funzionamento del programma.

2. CARICA DATI DA DISCO

Carica in memoria una serie di dati precedentemente memorizzati su disco. Viene richiesto il nome del file che deve essere presente sul disco inserito nel drive.
Viene indicato che è in corso il caricamento ed eventualmente che il file non è stato trovato. Per verificare il nome dei file presenti nel disco inserito sul drive è necessario uscire dal programma (con opzione n. 3 oppure con CONTROL + G) e dare il comando diretto <dir> quando ci si trova nella condizione COMMAND.

3. FINE PROGRAMMA

Permette di uscire dal programma perdendo però tutti i dati eventualmente presenti in memoria.

Tramite l'opzione numero 1 o 2 è possibile raggiungere il secondo menù che presenta 6 opzioni. Vediamole in dettaglio.

1. DISEGNO GRAFICO

Viene indicato il nome del file e quindi il grafico con accanto la legenda. Premendo il tasto <RETURN> si va al menù n. 2.

2. VEDERE I DATI

Stampa il nome del file, il numero progressivo, il valore dei dati mensili e il nome dei mesi corrispondenti.

3. CALCOLO DELLE PERCENTUALI

Opzione simile alla precedente ma con il calcolo delle percentuali.

4. CORREGGERE I DATI

Permette la correzione dei dati inseriti (da tastiera o da disco). Viene chiesto il numero dei dati da correggere (1/12), presentato il valore attuale del dato ed infine richiesto quello corretto.

5. SALVARE DATI SU DISCO

Salva su disco i dati inseriti o elaborati con il programma. Viene stampato il nome attuale del file e chiesto se si desidera cambiare il nome.

6. FINIRE

Ritorna al menù n. 1.

I dati generati con questo programma possono essere utilizzati direttamente dal programma n. 3 (diagramma a torta). Anche il programma <DIAGRAMMI A BARRE> può essere modificato in modo da poter stampare il grafico anche su carta per una maggiore portabilità delle informazioni. Altre modifiche attuabili al programma al fine di migliorarne la qualità possono interessare il numero massimo di dati inseribili, la possibilità di dare un nome agli stessi, la presentazione da programma dei file esistenti su disco eccetera. Sono comunque tutte modifiche apportabili con facilità data la realizzazione modulare del programma e la presenza di un'ampia documentazione. La grafica utilizza la massima risoluzione possibile ma i possessori di monitor a colori potranno inserire tutte le modifiche necessarie per una differenziazione cromatica delle barre stesse.

COME FUNZIONA

Il programma è costituito da una serie di routine autonome che eseguono una particolare funzione. La routine sicuramente più interes-

sante è quella che provvede alla rappresentazione grafica dei dati. Viene individuato il valore più grande ed in base ad esso calcolati tutti i valori corrispondenti. Ogni una di queste barre è riempita da un particolare disegno. Per motivi legati alla gestione dello schermo vengono stampate prima di tutto le scritte e quindi eseguiti i disegni.

Altra routine interessante riguarda la ricerca del file su disco. Si è fatto uso di una routine di errore per richiedere nuovamente il nome del file. Quando il computer cerca un file non presente su disco salta alla routine di errore (file non trovato nel drive indicato). La routine resetta la condizione di errore e richiede nuovamente il nome del file. Da tale condizione si può uscire solo con CONTROL + G.

COMMENTO SINGOLE ISTRUZIONI	
10/160	Inizializzazione delle variabili e caricamento dei valori delle DATA.
170/310	Presentazione del menù n. 1 e salto alle routine relativa.
320/580	Input dati da tastiera.
590/870	Input dati da disco.
890/990	Fine programma
1000/1210	Menù secondario.
1220/1470	Vedere i dati
1480/1780	Calcolo e stampa delle percentuali.
1790/1990	Correzione dei dati.
2000/2310	Salva dati su disco.
2320/2590	Legenda.
2600/2910	Disegno grafico.
2920/2960	DATA.

ELENCO VARIABILI			
pi	pigreco	i	ciclo caricamento dati
u	spazi	i$	lettura tastiera
nm	numero massimo dati	d$	opzione scelta
w	variabile di lavoro	fi	valore d$
n	numero attuale dati	s	somma dei valori
u$	scritte del menù	dc	numero dato da correggere
a()	matrice dei valori	o$	lettura tastiera
a$()	matrice dei nomi	xp	posizione di disegno
i%	ciclo di stampa	max	valore massimo
t$	lettura tastiera	yp	posizione di disegno
t	valore di t$	dy	posizione di disegno
p$	nome dei dati	hg	altezza relativa

```
10      rem *******************************
20      rem *                             *
30      rem *        PROGRAMMA n. 2       *
40      rem *                             *
50      rem *       DIAGRAMMA A BARRE     *
60      rem *                             *
70      rem *******************************
80      rem
90      fullw 2:clear
100     pi=3.1415927:u=26:nm=12:w=68:n=12
110     dim a(12),a$(12)
120     clearw 2:gotoxy 0,0
130     dim u$(10)
140     restore
150     for i%=1 to 12:a(i%)=0:read a$(i%):next
160     for i%=1 to 10:read u$(i%):next
170     ?:?:?
180     ? spc(u);"DIAGRAMMA A BARRE"
190     ?:?:?
200     close
210     ? spc(u);"1. Input dati da tastiera"
220     ?
230     ? spc(u);"2. Carica dati da disco"
```

```
240   ?
250   ? spc(u);"3. Fine programma"
260   ?:?:?:?:?:?
270   ? spc(u);
280   input "   Cosa scegli <1/3> ";t$
290   if t$<"1" or t$>"3" then 280
300   t=val(t$)
310   on t goto 320,590,880
320   rem
330   rem ***********************************
340   rem *                                 *
350   rem *       INPUT DATI DA TASTIERA     *
360   rem *                                 *
370   rem ***********************************
380   rem
390   clearw 2
400   gotoxy 0,2
410   ? spc (u);"INPUT DATI DA TASTIERA"
420   ?:?:?
430   input "                   Nome dei dati ";p$
440   if len(p$)=0 then ? "                  Ma de
ve avere un nome !!":?:goto 430
```

```
450   n=12

460   ?:?

470   for i=1 to n

480   ? "                              ";a$(i);tab(32);"
 = ";

490   input a(i)

500   a(i)=abs(a(i))

510   ? "                              ------------------------
----------------"

520   next

530   ?:?

540   ? "                                      FINE DEL CARIC
AMENTO"

550   ?

560   ? "                              Premi <RETURN> per
continuare...";

570   input "";i$

580   goto 1000

590   rem

600   rem ************************************

610   rem *                                  *

620   rem *        INPUT DATI DA DISCO       *

630   rem *                                  *

640   rem ************************************

650   rem
```

```
660   clearw 2

670   gotoxy 0,2

680   ? "                              CARICAMENTO
DATI DA DISCO"

690   ?:?:?

700   input "                              Nome dei
 dati ";p$

710   on error goto 840

720   if p$="" then 700

730   ?:?

740   ? "                              CARICAMENTO
IN CORSO....."

750   open "I",#1,p$

760   input #1,p$

770   for i%=1 to 12

780   input #1,a(i%)

790   next

800   close #1

810   clearw 2

820   gotoxy 0,2

830   goto 1000

840   ?:?

850   close #1

860   ? "                              --- File <";
p$;"> non trovato ---"
```

```
870   resume 690
880   rem
890   rem ************************************
900   rem *                                  *
910   rem *         FINE PROGRAMMA           *
920   rem *                                  *
930   rem ************************************
940   rem
950   clearw 2
960   gotoxy 0,0
970   fullw 3
980   clearw 3
990   end
1000  rem
1010  rem ************************************
1020  rem *                                  *
1030  rem *         MENU: SECONDARIO         *
1040  rem *                                  *
1050  rem ************************************
1060  rem
1070  clearw 2
1080  gotoxy 0,2
1090  ? spc(u);"OPZIONI DISPONIBILI"
```

```
1100  ?:?:?

1110  for i=1 to 10

1120  ? spc(u);u$(i)

1130  next

1140  ?:?

1150  ? "                              Scegli l:opz
ione <1/6> ";

1160  input "";d$

1170  if d$<"1" or d$>"6" then 1160

1180  fi=val(d$)

1190  clearw 2

1200  gotoxy 0,2

1210  on fi goto 2590,1220,1480,1790,2000,2280

1220  rem

1230  rem ***********************************

1240  rem *                                 *

1250  rem *         VEDERE I DATI           *

1260  rem *                                 *

1270  rem ***********************************

1280  rem

1290  clearw 2

1300  gotoxy 0,1

1310  ? "              DATI : ";p$

1320  ?:?
```

```
1330  ? spc(8);" n.";"                    dati";tab(
40);"  mese"
1340  ?
1350  for i%=1 to 12
1360  ? spc(8);
1370  ? using "###";i%;
1380  ? "     ";
1390  ? using "##########.###";a(i%);
1400  ? tab (40);a$(i%)
1410  next
1420  ?
1430  ? "                    Premi < RETURN > per c
ontinuare .....";
1440  input "";i$
1450  clearw 2
1460  gotoxy 0,2
1470  goto 1000
1480  rem
1490  rem ***********************************
1500  rem *                                 *
1510  rem *       CALCOLO PERCENTUALI       *
1520  rem *                                 *
1530  rem ***********************************
1540  rem
```

```
   1550  clearw 2

   1560  gotoxy 0,0

   1570  ? "                    CALCOLO PERCENTUALE
"

   1580  s=0

   1590  for i%=1 to n

   1600  s=s+a(i%)

   1610  next

   1620  ?:?

   1630  ? "          n.          Valore";tab(4
0);"Percentuale"

   1640  ?

   1650  for i=1 to n

   1660  ? "        ";

   1670  ? using "###";i;

   1680  ? "     ";

   1690  ? using "##########.###";a(i);

   1700  ? tab(40);:? using "###.###";100*a(i)/s;

   1710  ? "     ";a$(i)

   1720  next

   1730  ?:?

   1740  ? "                   Premi < RETURN >
per continuare ....";

   1750  input "";i$
```

```
1760  clearw 2
1770  gotoxy 0,2
1780  goto 1000
1790  rem
1800  rem ***********************************
1810  rem *                                 *
1820  rem *        CORREZIONE DATI          *
1830  rem *                                 *
1840  rem ***********************************
1850  rem
1860  clearw 2
1870  gotoxy 0,2
1880  ? "                    CORREZIONE DATI"
1890  ?:?:?
1900  ? "              Quale dato vuoi correggere
(1/12) : ";
1910  input "";dc
1920  if dc<1 or dc>12 then 1900
1930  ?
1940  ? "              Dato attuale : ";a(dc)
1950  input "              Dammi nuovo dato : ";a
(dc)
1960  goto 1970
1970  clearw 2
```

```
1980  gotoxy 0,2

1990  goto 1000

2000  rem

2010  rem ***********************************

2020  rem *                                 *

2030  rem *      SALVA DATI SU DISCO        *

2040  rem *                                 *

2050  rem ***********************************

2060  rem

2070  gotoxy 0,2

2080  ? "                         SALVATAGGIO DATI
 SU DISCO"

2090  ?:?:?

2100  close #1

2110  ?:?:?

2120  ? "                         Nome attuale fil
e : ";p$

2130  ?:? "                         Vuoi cambiare
<s/n> ";

2140  input "";o$

2150  if o$="S" or o$="s" then 2160 else 2170

2160  input "                         Nome nuovo :
 ";p$

2170  ?:?:?
```

```
  2180  ? "                          SALVATAGGIO IN C
ORSO"
  2190  ?:?
  2200  open "O",#1,p$
  2210  ? #1,p$
  2220  for i%=1 to 12
  2230  ? #1,a(i%)
  2240  next
  2250  clearw 2
  2260  gotoxy 0,2
  2270  goto 1000
  2280  clearw 2
  2290  gotoxy 0,2
  2300  erase a,a$,u$
  2310  goto 90
  2320  rem
  2330  rem ****************************
  2340  rem *                          *
  2350  rem *          LEGENDA         *
  2360  rem *                          *
  2370  rem ****************************
  2380  rem
  2390  xp=340
```

```
2400  ? "    File : ";p$
2410  ?:?:?
2420  for i%=1 to 12
2430  ? tab(53);
2440  ? using "##";i%;
2450  ? ". ";a$(i%)
2460  next
2470  ?:?
2480  ? "    Premi < RETURN > per continuare ..
.. ";
2490  linef xp,w+2,60+xp,2+w
2500  linef 60+xp,2+w,60+xp,342-w
2510  linef xp,w+2,xp,342-w
2520  linef xp,342-w,60+xp,342-w
2530  for i=1 to 12
2540  linef xp,i*17+2+w,60+xp,i*17+2+w
2550  color 1,1,1,i,2
2560  fill xp+4,i*17+w
2570  next
2580  return
2590  rem
2600  rem **************************
2610  rem *                        *
2620  rem *     DISEGNO GRAFICO    *
```

```
2630  rem *                         *
2640  rem **************************
2650  rem
2660  clearw 2
2670  gotoxy 0,0
2680  gosub 2320
2690  max=0
2700  for i%=1 to 12
2710  if a(i%)>max then max=a(i%)
2720  next
2730  yp=300
2740  dy=250
2750  for i%=1 to n
2760  color 1,1,1,i%,2
2770  xp=(i%-1)*24+(320-(n*24))/2
2780  hg=250*a(i%)/max
2790  linef xp,yp,xp,yp-hg
2800  linef xp,yp-hg,xp+19,yp-hg
2810  linef xp+19,yp-hg,xp+19,yp
2820  linef xp+19,yp,xp,yp
2830  linef xp,yp-hg,xp+4,yp-hg-4
2840  linef xp+4,yp-hg-4,xp+23,yp-hg-4
2850  linef xp+19,yp-hg,xp+23,yp-hg-4
```

```
 2860  linef xp+23,yp-hg-4,xp+23,yp-4

 2870  linef xp+23,yp-4,xp+19,yp

 2880  if hg>2 then fill xp+2,yp-1

 2890  next

 2900  input " ";i$

 2910  goto 1000

 2920  data GENNAIO,FEBBRAIO,MARZO,APRILE,MAGGIO
,GIUGNO,LUGLIO

 2930  data AGOSTO,SETTEMBRE,OTTOBRE,NOVEMBRE,DI
CEMBRE

 2940  data 1. Disegno grafico,2. Vedere i dati

 2950  data 3. Calcolo delle percentuali,4. Corr
eggere i dati

 2960  data 5. Salvare dati su disco,6. Finire,,
,,
```

Programma n.3

TITOLO : DIAGRAMMA A TORTA
LUNGHEZZA : 8347 Byte

DESCRIZIONE:

Analogamente al programma precedente, quello proposto è in grado di rappresentare in forma grafica una serie di valori (anche non correlati tra loro) sotto forma di grafico a torta.
Il principio di funzionamento è del tutto simile al precedente anche se l'output è profondamente diverso. Si tratta di dare alla somma dei valori l'interezza del cerchio (360 gradi) e quindi di ripartire in proporzione i singoli valori. Data la somiglianza di questo programma con quello proposto in precedenza vi rimando al programma n. 2 per maggiori dettagli sul funzionamento e sui comandi da impartire. È da notare l'assoluta compatibilità dei dati tra i due programmi che potranno essere quindi rappresentati sia sotto forma di barre che di diagramma a torta.

COME FUNZIONA:

Si è gia avuto modo di dire che il il programma sfrutta un algoritmo molto semplice che riportiamo di seguito per consentire a chi lo voglia di modificare il programma a proprio piacimento. Dati una serie di 12 valori mensili (ad esempio ricavi) viene eseguita la somma e quindi applicata la seguente formula:

$$An = S*Vx/360$$

dove: An = angolo calcolato
S = somma dei valori
Vx = Valore x

Dopo aver calcolato i valori e averli tradotti in linee che suddividono il cerchio è neccessario colorare l'interno. A tal fine si utilizza il comando FILL seguito dalle coordinate e dal tipo di retinatura voluta. È stato neccessario dotare il programma di una protezione nei confronti di quelle aree (e quindi di valori) particolarmente piccole che frequentemente danno problemi in fase di riempimento.

COMMENTO SINGOLE ISTRUZIONI	
10/160	Inizializzazione delle variabili e caricamento dei valori dalle DATA.
170/310	Presentazione del menù n. 1 e salto alle routine relativa.
320/580	Input dati da tastiera.
590/870	Input dati da disco.
890/990	Fine programma.
1000/1210	Menù secondario.
1220/1470	Vedere i dati.
1480/1780	Calcolo e stampa delle percentuali.
1790/1990	Correzione dei dati.
2000/2310	Salva dati su disco.
2320/2580	Legenda.
2590/2840	Disegno grafico.
2850/2890	DATA.

ELENCO VARIABILI			
pi	pigreco	i$	lettura tastiera
u	spazi	d$	opzione scelta
nm	numero massimo dati	fi	valore d$
w	variabile di lavoro	s	somma dei valori
n	numero attuale dati	dc	numero dato da correggere
u$	scritte del menù	o$	lettura tastiera
a()	matrice dei valori	xp	posizione di disegno
a$()	matrice dei nomi	max	valore massimo
i%	ciclo di stampa	yp	posizione di disegno
t$	lettura tastiera	dy	posizione di disegno
t	valore di t$	hg	altezza relativa
p$	nome dei dati	at	angolo totale
i	ciclo caricamento dati	an	angolo attuale

```
10    rem *********************************
20    rem *                               *
30    rem *         PROGRAMMA n. 3        *
40    rem *                               *
50    rem *       DIAGRAMMA A TORTA       *
60    rem *                               *
70    rem *********************************
80    rem
90    fullw 2:clear
100   pi=3.1415927:u=26:nm=12:w=68:n=12
110   dim a(12),a$(12)
120   clearw 2:gotoxy 0,0
130   dim u$(10)
140   restore
150   for i%=1 to 12:a(i%)=0:read a$(i%):next
160   for i%=1 to 10:read u$(i%):next
170   ?:?:?
180   ? spc(u);"DIAGRAMMA A TORTA"
190   ?:?:?
200   close
210   ? spc(u);"1. Input dati da tastiera"
220   ?
230   ? spc(u);"2. Carica dati da disco"
```

```
240   ?
250   ? spc(u);"3. Fine programma"
260   ?:?:?:?:?:?
270   ? spc(u);
280   input "   Cosa scegli <1/3> ";t$
290   if t$<"1" or t$>"3" then 280
300   t=val(t$)
310   on t goto 320,590,880
320   rem
330   rem ***********************************
340   rem *                                 *
350   rem *      INPUT DATI DA TASTIERA     *
360   rem *                                 *
370   rem ***********************************
380   rem
390   clearw 2
400   gotoxy 0,2
410   ? spc (u);"INPUT DATI DA TASTIERA"
420   ?:?:?
430   input "                 Nome dei dati ";p$
440   if len(p$)=0 then ? "                 Ma de
ve avere un nome !!":?:goto 430
450   n=12
```

```
460   ?:?
470   for i=1 to n
480   ? "                          ";a$(i);tab(32);"
 = ";
490   input a(i)
500   a(i)=abs(a(i))
510   ? "                          ------------------
-----------"
520   next
530   ?:?
540   ? "                          FINE DEL CARIC
AMENTO"
550   ?
560   ? "                      Premi <RETURN> per
continuare...";
570   input "";i$
580   goto 1000
590   rem
600   rem ***********************************
610   rem *                                 *
620   rem *       INPUT DATI DA DISCO       *
630   rem *                                 *
640   rem ***********************************
650   rem
660   clearw 2
```

```
670   gotoxy 0,2

680   ? "                              CARICAMENTO
DATI DA DISCO"

690   ?:?:?

700   input "                              Nome dei
 dati ";p$

710   on error goto 840

720   if p$="" then 700

730   ?:?

740   ? "                              CARICAMENTO
IN CORSO....."

750   open "I",#1,p$

760   input #1,p$

770   for i%=1 to 12

780   input #1,a(i%)

790   next

800   close #1

810   clearw 2

820   gotoxy 0,2

830   goto 1000

840   ?:?

850   close #1

860   ? "                              --- File <";
p$;"> non trovato ---"
```

```
870   resume 690
880   rem
890   rem *************************************
900   rem *                                   *
910   rem *         FINE PROGRAMMA            *
920   rem *                                   *
930   rem *************************************
940   rem
950   clearw 2
960   gotoxy 0,0
970   fullw 3
980   clearw 3
990   end
1000  rem
1010  rem *************************************
1020  rem *                                   *
1030  rem *         MENU: SECONDARIO          *
1040  rem *                                   *
1050  rem *************************************
1060  rem
1070  clearw 2
1080  gotoxy 0,2
1090  ? spc(u);"OPZIONI DISPONIBILI"
```

```
1100  ?:?:?
1110  for i=1 to 10
1120  ? spc(u);u$(i)
1130  next
1140  ?:?
1150  ? "                          Scegli l:opz
ione <1/6> ";
1160  input "";d$
1170  if d$<"1" or d$>"6" then 1160
1180  fi=val(d$)
1190  clearw 2
1200  gotoxy 0,2
1210  on fi goto 2590,1220,1480,1790,2000,2280
1220  rem
1230  rem ***********************************
1240  rem *                                 *
1250  rem *         VEDERE I DATI           *
1260  rem *                                 *
1270  rem ***********************************
1280  rem
1290  clearw 2
1300  gotoxy 0,1
1310  ? "             DATI : ";p$
1320  ?:?
```

```
1330  ? spc(8);" n.";"                    dati";tab(
40);"  mese"

1340  ?

1350  for i%=1 to 12

1360  ? spc(8);

1370  ? using "###";i%;

1380  ? "      ";

1390  ? using "##########.###";a(i%);

1400  ? tab (40);a$(i%)

1410  next

1420  ?

1430  ? "                    Premi < RETURN > per c
ontinuare .....";

1440  input "";i$

1450  clearw 2

1460  gotoxy 0,2

1470  goto 1000

1480  rem

1490  rem ************************************

1500  rem *                                  *

1510  rem *        CALCOLO PERCENTUALI       *

1520  rem *                                  *

1530  rem ************************************

1540  rem
```

```
1550  clearw 2
1560  gotoxy 0,0
1570  ? "                    CALCOLO PERCENTUALE
"
1580  s=0
1590  for i%=1 to n
1600  s=s+a(i%)
1610  next
1620  ?:?
1630  ? "            n.             Valore";tab(4
0);"Percentuale"
1640  ?
1650  for i=1 to n
1660  ? "          ";
1670  ? using "###";i;
1680  ? "      ";
1690  ? using "##########.###";a(i);
1700  ? tab(40);:? using "###.###";100*a(i)/s;
1710  ? "     ";a$(i)
1720  next
1730  ?:?
1740  ? "                     Premi < RETURN > p
er continuare ....";
1750  input "";i$
1760  clearw 2
```

```
1770  gotoxy 0,2

1780  goto 1000

1790  rem

1800  rem ************************************

1810  rem *                                  *

1820  rem *        CORREZIONE DATI           *

1830  rem *                                  *

1840  rem ************************************

1850  rem

1860  clearw 2

1870  gotoxy 0,2

1880  ? "                         CORREZIONE DATI"

1890  ?:?:?

1900  ? "               Quale dato vuoi correggere
 (1/12) : ";

1910  input "";dc

1920  if dc<1 or dc>12 then 1900

1930  ?

1940  ? "               Dato attuale : ";a(dc)

1950  input "                Dammi nuovo dato : ";a
(dc)

1960  goto 1970

1970  clearw 2

1980  gotoxy 0,2
```

```
1990  goto 1000

2000  rem

2010  rem ***********************************

2020  rem *                                 *

2030  rem *      SALVA DATI SU DISCO        *

2040  rem *                                 *

2050  rem ***********************************

2060  rem

2070  gotoxy 0,2

2080  ? "                    SALVATAGGIO DATI
 SU DISCO"

2090  ?:?:?

2100  close #1

2110  ?:?:?

2120  ? "                    Nome attuale fil
e : ";p$

2130  ?:? "                   Vuoi cambiare
<s/n> ";

2140  input "";o$

2150  if o$="S" or o$="s" then 2160 else 2170

2160  input "                    Nome nuovo :
 ";p$

2170  ?:?:?

2180  ? "                    SALVATAGGIO IN C
ORSO"

2190  ?:?
```

```
2200  open "O",#1,p$
2210  ? #1,p$
2220  for i%=1 to 12
2230  ? #1,a(i%)
2240  next
2250  clearw 2
2260  gotoxy 0,2
2270  goto 1000
2280  clearw 2
2290  gotoxy 0,2
2300  erase a,a$,u$
2310  goto 90
2320  rem
2330  rem ***************************
2340  rem *                         *
2350  rem *         LEGENDA         *
2360  rem *                         *
2370  rem ***************************
2380  rem
2390  xp=340
2400  ? "    File : ";p$
2410  ?:?:?
2420  for i%=1 to 12
```

```
2430  ? tab(53);
2440  ? using "##";i%;
2450  ? ".  ";a$(i%)
2460  next
2470  ?:?
2480  ? "     Premi < RETURN > per continuare ..
.. ";
2490  linef xp,w+2,60+xp,2+w
2500  linef 60+xp,2+w,60+xp,342-w
2510  linef xp,w+2,xp,342-w
2520  linef xp,342-w,60+xp,342-w
2530  for i=1 to 12
2540  linef xp,i*17+2+w,60+xp,i*17+2+w
2550  color 1,1,1,i,2
2560  fill xp+4,i*17+w
2570  next
2580  return
2590  rem
2600  rem **************************
2610  rem *                        *
2620  rem *     DISEGNO GRAFICO    *
2630  rem *                        *
2640  rem **************************
2650  rem
```

```
2660  clearw 2
2670  gotoxy 0,0
2680  gosub 2320
2690  circle 170,170,130
2700  s=0
2710  for i%=1 to n
2720  s=s+a(i%)
2730  next
2740  linef 170,170,170,40
2750  at=0
2760  for i%=1 to 12
2770  color 1,1,1,i%,2
2780  if s<>0 then an=2*pi*a(i%)/s
2790  at=at+an
2800  linef 170,170,170+130*sin(at),170-130*cos
(at)
2810  if an>0.05 then fill 170+126*sin(at-an/2)
,170-126*cos(at-an/2)
2820  next
2830  input " ";i$
2840  goto 1000
2850  data GENNAIO,FEBBRAIO,MARZO,APRILE,MAGGIO
,GIUGNO,LUGLIO
2860  data AGOSTO,SETTEMBRE,OTTOBRE,NOVEMBRE,DI
CEMBRE
```

```
   2870  data 1. Disegno grafico,2. Vedere i dati

   2880  data 3. Calcolo delle percentuali,4. Corr
eggere i dati

   2890  data 5. Salvare dati su disco,6. Finire,,
,,
```

Programma n.4

TITOLO : CALCOLO AREE CON GAUSS
LUNGHEZZA : 2579 Byte

DESCRIZIONE:

Il programma calcola l'area di una figura definita dalle coordinate cartesiane dei suoi vertici. Si possono inserire figure che hanno un minimo di 3 vertici ad un massimo di 20.

Questo limite è assolutamente superabile modificando qualche linea di programma e tiene conto che normalmente non ci si trova nelle condizioni di avere aree così estese. L'utilizzo trova ampia applicazione in topografia, nel calcolo di aree reagenti in ingegneria, statica e nel calcolo di qualche integrale definito e dalla precisione non eccezionale. Il programma richiede il numero di vertici da inserire e quindi i valori delle relative coordinate i cui campi di variabilità sono compresi tra 0 e 260. I vertici dovranno essere inseriti in maniera ordinata percorrendo la figura da sinistra a destra. Una volta conclusa la fase di inserimento dei dati viene chiesto di premere un tasto e quindi disegnata sullo schermo la figura ottenuta dalla connessione dei vari vertici. Sulla destra dello schermo compaiono i valori delle coordinate. Questo è uno dei motivi che hanno indotto a limitare a 20 il numero dei vertici da inserire. Sul disegno i vertici sono evidenziati da alcuni cerchietti. Nella parte bassa è indicata l'area della figura priva di unità di misura e quindi utilizzabile in tutte le occasioni. Sia i valori delle coordinate che il risulatato del calcolo hanno una precisione di due cifre dopo la virgola.
Le modifiche inseribili riguardano la possibilità di salvare su disco i valori digitati, di stampare sia il disegno che le coordinate e di aumentare il numero totale di vertici. Per quanto riguarda la possibilità di salvare i dati su disco si potrà fare riferimento ad uno dei programmi precedenti che prevedono pure la possibilità di gestire gli errori. Aumentando il numero totale di vertici (e dovendo ricorrere ad una nuova pagina per la loro rappresentazione numerica) si potranno aumentare anche le dimensioni della pagina grafica e modificare anche i valori massimi delle stesse coordinate.

COMMENTO SINGOLE ISTRUZIONI	
10/420	Inizialializzazione variabili e input dati.
430/690	Calcolo area.
700/950	Stampa disegno e valori coordinate.

ELENCO VARIABILI	
x()	Matrice ascisse
y()	Matrice ordinate
p$	Stringa informativa
i%	Ciclo caricamento stringa
b$	Stringa costituita
n	Numero coordinate da inserire
i	Ciclo di input dati
q$	Lettura tastiera
s	Area calcolata
i$	Lettura tastiera

```
10   rem ***********************************
20   rem *                                 *
30   rem *         PROGRAMMA n. 4          *
40   rem *                                 *
50   rem * CALCOLO AREA CON COORDINATE     *
60   rem * CARTESIANE : Formule di Gauss   *
70   rem *                                 *
80   rem ***********************************
90   rem
100  dim x(41)
110  dim y(41)
120  p$="Premi un <RETURN> per continuare... "
130  for i%=1 to 40
140  b$=b$+"-"
150  next
160  fullw 2
```

```
170   clearw 2
180   gotoxy 11,3
190   ? "CALCOLO AREA POLIGONO CON GAUSS"
200   gotoxy 11,4
210   ? "--------------------------------"
220   gotoxy 9,7
230   input "Quante coordinate vuoi inserire (3
/20) ";n
240   n=int(abs(n))
250   if n<3 or n>20 then 180
260   ?:?
270   for i=1 to n
280   ? tab(20);"x(";i;
290   ? ")  <0/260>  ";
300   input x(i)
310   if x(i)<0 or x(i)>300 then 280
320   ? tab(20);"y(";i;
330   ? ")  <0/260>  ";
340   input y(i)
350   if y(i)<0 or y(i)>300 then 320
360   ? tab(20);b$
370   next
380   ?:?
```

```
390    ? tab(20);b$
400    ? tab(20);p$;
410    input q$
420    clearw 2
430    rem
440    rem      ******************************
450    rem      *                            *
460    rem      *          CALCOLO           *
470    rem      *                            *
480    rem      ******************************
490    rem
500    s=0
510    y(0)=y(n)
520    y(n+1)=y(1)
530    for i=1 to n
540    s=s+x(i)*(y(i-1)-y(i+1))
550    next
560    s=s/2
570    gotoxy 0,0
580    ? "            CALCOLO AREA CON GAUSS";
590    for i=1 to n
600    if i>1 then ?
610    ? tab(42);"x(";
```

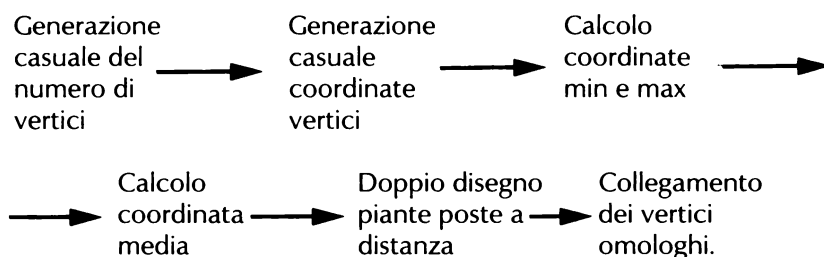
```
620   ? using "##";i;
630   ? ") =";
640   ? using "###.##";x(i);
650   ? "    y(";
660   ? using "##";i;
670   ? ") =";
680   ? using "###.##";y(i);
690   next
700   rem
710   rem    ****************************
720   rem    *                          *
730   rem    *         DISEGNO          *
740   rem    *                          *
750   rem    ****************************
760   rem
770   linef 40,314,300,314
780   linef 40,314,40,24
790   linef 40,24,300,24
800   linef 300,24,300,314
810   for i=1 to n-1
820   linef x(i)+40,314-y(i),x(i+1)+40,314-y(i+
1)
830   next
```

```
840    linef x(1)+40,314-y(1),x(n)+40,314-y(n)
850    color 1,1,1
860    for i=1 to n
870    pcircle x(i)+40,314-y(i),3
880    next
890    gotoxy 0,18
900    ? "   Area = ";
910    ? using "#####.##";abs(s);
920    ? "  <RETURN> ";
930    input i$
940    clearw 2
950    goto 160
```

Programma n. 5

TITOLO : GENERATORE FIGURE TRIDIMENSIONALI
LUNGHEZZA : 937 Byte

DESCRIZIONE:

Il programma genera una serie casuale di figure tridimensionali.
È un programma dimostrativo ma che al suo interno contiene tutti gli elementi per realizzare delle immagini in tre dimensioni. Il principio di funzionamento è semplice: si tratta di disegnare due piante (ottenute dalla generazione casuale dei vertici) ad una certa distanza l'una dall'altra e quindi collegarne i vertici omologhi. Dopo la generazione delle coordinate viene trovato il punto medio della figura stessa al fine di porla costantemente al centro dello schermo. Lo schema seguente servirà a capire meglio questo principio che viene usato molto spesso nelle rappresentazioni col computer di oggetti reali in assonometria.

Il programma una volta terminata una generazione ricomincerà tutto da capo. Per uscire dal programma sarà neccessario premere contemporaneamente i tasti CONTROL + G.

COMMENTO SINGOLE ISTRUZIONI	
10/180	Inizializzazione del programma, generazione casuale del numero di vertici, coordinate e riempimento delle relative matrici.
190	Calcolo coordinata inizio del disegno.
200/250	Disegno delle linee tra i vertici.
260/290	Chiusura della figura.

ELENCO VARIABILI	
n	Numero totale di vertici
i%	Ciclo generazione casuale vertici
a()	Matrice delle ascisse
b()	Matrice delle ordinate
max	Valore ascissa massima
min	Valore ordinata minima
i%	Ciclo disegno
xp	Coordinata media

```
10    rem    ***************************
20    rem    *                         *
30    rem    *      PROGRAMMA n. 5     *
40    rem    *                         *
50    rem    *   GENERATORE FIGURE     *
60    rem    *    TRIDIMENSIONALI      *
70    rem    *                         *
80    rem    ***************************
90    rem
100   fullw 2
110   clearw 2
120   n=rnd(1)*7+4
130   for i%=1 to n
140   a(i%)=rnd(1)*300
150   b(i%)=rnd(1)*150
160   if a(i%)>max then max=a(i%)
170   if b(i%)<min then min=b(i%)
```

```
   180    next

   190    xp=(614-max+min)/2

   200    clearw 2

   210    for i%=1 to n-1

   220    linef a(i%)+xp,300-b(i%),a(i%+1)+xp,300-b
(i%+1)

   230    linef a(i%)+xp,160-b(i%),a(i%+1)+xp,160-b
(i%+1)

   240    linef a(i%)+xp,300-b(i%),a(i%)+xp,160-b(i
%)

   250    next

   260    linef a(1)+xp,300-b(1),a(n)+xp,300-b(n)

   270    linef a(1)+xp,160-b(1),a(n)+xp,160-b(n)

   280    linef a(n)+xp,300-b(n),a(n)+xp,160-b(n)

   290    goto 120
```

Programma n. 6

TITOLO : BILANCIO ECONOMICO
LUNGHEZZA : 8298 Byte

DESCRIZIONE:

Il programma permette la gestione di un piccolo bilancio economico per mezzo di una serie di particolari istruzioni tutte tramite menù e quindi facilmente utilizzabili anche dai meno esperti nel settore. Il programma inizia con la presentazione del titolo e la richiesta di premere il tasto di <RETURN>. A questo punto si passa ad una serie di opzioni che vediamo in dettaglio.

1. INSERIMENTO DATI

Viene proposto il numero dell'inserimento, il giorno, mese e anno della registrazione da effettuare. Su tali valori non viene eseguito un controllo in quanto si possono forzare date volontariamente errate per un particolare tipo di record da riconoscere di seguito.
Si richiede quindi il valore (in positivo se ENTRATA e in negativo se USCITA). Questo valore può essere seguito da una breve causale. Dopo l'inserimento il programma chiede se si vuole continuare (s/n). In caso di risposta affermativa il programma continua con il record successivo altrimenti ritorna alla fase di menù.

2. VISUALIZZAZIONE DATI

Riporta sullo schermo il numero progressivo della registrazione, la data (gg/mm/aaaa), l'entrata o l'uscita e la causale. La pressione del tasto di <RETURN> riconduce al menù principale.

3. CARICAMENTO DATI DA DISCO

Prima di effettuare questa opzione bisogna inserire nel drive un disco precedentemente inizializzato da questo programma e contenente quindi un file compatibile. Il programma avvertirà di attendere e quindi effettuerà il caricamento dei dati perdendo quelli eventualmente presenti in memoria. Il nome dato al file è < BILANCIO > ma può essere facilmente modificato o reso modificabile da programma. Nel caso il disco fosse sbagliato o il file non creato il programma avvertirà con la scritta <FILE NON ESISTENTE !!> ritornando quindi alle opzioni alla pressione del tasto <RETURN>.

4. CALCOLO TOTALI

Effettua il calcolo totale separato delle entrate, delle uscite e il consuntivo. Nel testo appare la dicitura £ ma tale unità monetaria si può modificare da programma (es. si può inserire $). Come sempre <RETURN> riporta alle opzioni.

5. CORREZIONE / CANCELLAZIONE DEI DATI

L'attivazione di questo comando porta ad un nuovo menù interno al precedente. Le opzioni disponibili sono tre; vediamole in ordine.

A. CANCELLAZIONE
Chiede il numero del RECORD da cancellare compreso tra 1 e il numero massimo dei RECORD inseriti.

B. CORREZIONE
Come per il precedente si richiede il numero del RECORD da correggere. Se l'input è esatto (se quindi la risposta è compresa tra 1 e il numero massimo di dati inseriti) viene proposto il valore attuale e richiesto quello corretto.

C. RITORNO MENÙ
Permette di uscire dall'opzione senza apportare correzioni o cancellazioni.

6. SALVATAGGIO DATI SU DISCO

Il disco non deve essere fisicamente protetto e inserito nel drive prima di dare il comando. Qualora questo non si verificasse il sistema chiede se si vuole cancellare la richiesta oppure riprovare. Nel nostro caso sarà utile inserire un disco e quindi puntare il mouse su <RETRY>. Si fa notare che avendo il file sempre lo stesso nome un eventuale file di questo tipo presente nel disco verrebbe cancellato. Il programma è stato impostato in questo modo perchè di solito la prima operazione da effettuare quando si usa il programma è quella si richiamare il file esistente e aggiungere a questo i nuovi dati. Se si intende usare sempre lo stesso disco per varie registrazioni è neccessario modificare il programma in modo che chieda il nome del file all'inizio e lo utilizzi poi in tutte le fasi di input / output.

7. FINE PROGRAMMA

Chiede se si intende realmente abbandonare il programma e in caso di risposta affermativa ritorna al BASIC. Prima di attivare questo comando i dati eventualmente creati o modificati dovranno essere salvati altrimenti andranno irrimediabilmente persi.

Le modifiche attuabili riguardano la gia citata possibilità di cambiare il nome del file di dati, l'output (peraltro utilissimo) sulla stampante, e la possibilità di correggere non solo il valore del RECORD ma anche la data e la causale.

COMMENTO SINGOLE ISTRUZIONI	
10/290	Inizializzazione del programma e delle variabili, presentazione.
300/360	Menù principale e salto alla routine relativa.
370/770	Inserimento dei dati.
780/990	Visualizzazione dei dati.
1000/1260	Caricamento dati da disco.
1270/1560	Calcolo totali.
1570/2020	Correzione e cancellazione dei dati.
2030/2280	Salvataggio dati su disco.
2290/2420	Fine programma.
2430/2590	Routine di errore del disco.

ELENCO VARIABILI	
nm	Numero massimo di dati
n$	Nome del file
d$()	Matrice data
a$()	Matrice della causale
c()	Matrice del capitale
i$	Lettura tastiera
n	Numero puntato del RECORD
g$	Numero del giorno
m$	Numero del mese
a$	Numero dell'anno
v$	Valore
l	Lunghezza di v$
s	Ciclo controllo input valore
h$	Stringa parziale
te	Totale entrate
tu	Totale uscite
t	Totale
nc	Numero record da cancellare o correggere

```
10     rem *****************************
20     rem *                           *
30     rem *       PROGRAMMA n. 6      *
40     rem *                           *
50     rem *     BILANCIO ECONOMICO    *
60     rem *                           *
70     rem *****************************
80     rem
90     fullw 2
100    clearw 2
110    nm=255        : rem   numero massimo dat
i
120    n$="bilancio" : rem   nome file
130    dim a$(nm),d$(nm),c(nm)
140    gotoxy 0,0
150    ?:?:?:?:?
160    ? "                          BILANCIO
 ECONOMICO"
170    ?:?:?:?:?:?
180    ? "                          Premi <R
ETURN> ..";
190    input i$
200    clearw 2
210    gotoxy 0,0
```

```
  220   ?:?:?

  230   ? "                               ----------<
 OPZIONI  >---------"

  240   ?:?:?

  250   ? "                               1.  INSERIMEN
TO DATI"

  260   ? "                               2.  VISUALIZZ
AZIONE DATI"

  270   ? "                               3.  CARICAMEN
TO DATI DA DISCO"

  280   ? "                               4.  CALCOLO T
OTALI"

  290   ? "                               5.  CORREZION
E / CANCELLAZIONE DATI"


  300   ? "                               6.  SALVATAGG
IO DATI SU DISCO"

  310   ? "                               7.  FINE PROG
RAMMA"

  320   ?:?:?

  330   ? "                                  PREMI UN
NUMERO <1/7> .. ";

  340   input i$

  350   if i$<"1" or i$>"7" then 200

  360   on val(i$) goto 370,780,1000,1270,1570,20
30,2290

  370   rem

  380   rem *****************************
```

```
390   rem *                          *
400   rem *     INSERIMENTO DATI      *
410   rem *                          *
420   rem ****************************
430   rem
440   clearw 2
450   gotoxy 0,0
460   ?:?:?
470   n=n+1
480   if n>nm then ?:?:?:?:? "
   TROPPI RECORD":goto 730


490   ? "                    ----------   INP
UT DATI  ----------"

500   ? "                    Record n. ";n

510   ?

520   input "                        Giorno
: ";g$

530   l=len(g$):if l<1 or l>2 then 520

540   if l=1 then g$="0"+g$

550   input "                        Mese
: ";m$

560   l=len(m$):if l<1 or l>2 then 550

570   if l=1 then m$="0"+m$
```

```
580   input "                        Anno
: ";q$

590   l=len(q$):if l<2 or l>4 then 580

600   if l=2 then q$="19"+q$

610   if l=3 then q$="1"+q$

620   d$(n)=g$+"/"+m$+"/"+q$

630   ?

640   input "                        Valore (+/-)
: ";v$

650   l=len(v$):if l<1 or l>9 then 640

660   for s=1 to l

670   h$=mid$(v$,s,1):if h$="-" or h$="+" and s
=1 then 690

680   if h$<"0" or h$>"9" then 640

690   next

700   c(n)=val(v$)

710   ?

720   input "                        Casuale
: ";a$(n)

730   ?:?:?

740   input "                        Ancora <s/n>
.. ";i$

750   if i$="n" or i$="N" then 200 else 440

760   input "                        Premi <RETURN
> .. ";i$

770   goto 200
```

```
780   rem
790   rem *****************************
800   rem *                           *
810   rem *  VISUALIZZAZIONE DATI     *
820   rem *                           *
830   rem *****************************
840   rem
850   clearw 2
860   gotoxy 0,0
870   ? "  n.     data          ENTRATE          USCI
TE     casuale "
880   ? "--------------------------------------------------
--------------------------------"
890   for i=1 to n
900   ? using "###";i;
910   ? "  ";d$(i);
920   ? tab(18);
930   if c(i)>0 then ? using "##########";c(i);
: ? "   ----------";
940   if c(i)<0 then ? "----------    ";:? using
 "##########";-c(i);
950   ? tab(45);a$(i)
960   next
970   ?:?
```

```
980   input "                      Premi <RETURN>
.. ";i$
990   goto 200
1000  rem
1010  rem ******************************
1020  rem *                            *
1030  rem * CARICAMENTO DATI DA DISCO *
1040  rem *                            *
1050  rem ******************************
1060  rem
1070  clearw 2
1080  gotoxy 0,0
1090  ?:?:?:?
1100  ? "                      CARICAMENTO
 DATI DA DISCO"
1110  ?:?:?:?
1120  ? "                      *******    A
TTENDI  ******"
1130  on error goto 2430
1140  open "I",#1,n$
1150  input #1,n
1160  for i%=1 to n
1170  input #1,a$(i%)
1180  input #1,d$(i%)
1190  input #1,c(i%)
```

```
1200  next

1210  close

1220  ?:?:?:?

1230  ? "                                    CARICAMEN
TO EFFETTUATO"

1240  ?:?

1250  input "                                 Premi <
RETURN> ... ";i$

1260  goto 200

1270  rem

1280  rem ****************************

1290  rem *                          *

1300  rem *      CALCOLO TOTALI      *

1310  rem *                          *

1320  rem ****************************

1330  rem

1340  clearw 2

1350  gotoxy 0,0

1360  te=0:tu=0:t=0

1370  for i%=1 to nm

1380  if c(i%)>0 then te=te+c(i%)

1390  if c(i%)<0 then tu=tu-c(i%)

1400  t=t+c(i%)

1410  next
```

```
1420  gotoxy 0,0
1430  ?:?:?:?:?
1440  ? "                                   CALC
OLO TOTALI"
1450  ?:?:?
1460  ? "                 Totale entrate : | ";
1470  ? using "############";te
1480  ? "                 Totale uscite  : | ";
1490  ? using "############";tu
1500  ?
1510  ? "                 Totale         : | ";
1520  ? using "############";t
1530  ?:?:?
1540  ? "                                   Prem
i <RETURN> .. ";
1550  input i$
1560  goto 200
1570  rem
1580  rem *********************************
1590  rem *                               *
1600  rem * CORREZIONE/CANCELLAZIONE DATI *
1610  rem *                               *
1620  rem *********************************
1630  rem
```

```
   1640  clearw 2

   1650  gotoxy 0,0

   1660  ?:?:?

   1670  ?  "                        CANCELLAZIONE
 / CORREZIONE"

   1680  ?:?

   1690  ?  "                        1. CANCELLAZI
ONE"

   1700  ?  "                        2. CORREZIONE
"

   1710  ?  "                        3. RITORNO ME
NU:"

   1720  ?:?

   1730  input "                          Cosa sceg
li <1/3> .. ";i$

   1740  if i$<"1" or i$>"3" then 1640

   1750  on val(i$) goto 1760,1910,200

   1760  clearw 2 : rem cancellazione

   1770  gotoxy 0,0

   1780  ?:?:?:?

   1790  ?  "                          CANCELLAZI
ONE"

   1800  ?:?:?

   1810  ?  "               Quale RECORD vuoi canc
ellare < 1 /";n;" > ";

   1820  input nc
```

```
1830  if nc<1 or nc>n then 1810
1840  for f%=nc+1 to n
1850  a$(f%-1)=a$(f%)
1860  d$(f%-1)=d$(f%)
1870  c(f%-1)=c(f%)
1880  next
1890  n=n-1
1900  goto 200
1910  clearw 2 : rem correzione
1920  gotoxy 0,0
1930  ?:?:?:?
1940  ? "                         CORREZIONE
"
1950  ?:?:?
1960  ?  "                   Quale RECORD vuoi co
rreggere < 1 /";n;">";
1970  input nc
1980  ?:?:?:?
1990  ? "                   Valore attuale  ";c(
nc)
2000  ?:?
2010  input "                   Valore corretto
";c(nc)
2020  goto 200
```

```
2030  rem
2040  rem ****************************
2050  rem *                          *
2060  rem * SALVATAGGIO DATI SU DISCO *
2070  rem *                          *
2080  rem ****************************
2090  rem
2100  clearw 2
2110  gotoxy 0,0
2120  ?:?:?:?
2130  ? "                              CARICAMENTO
 DATI SU DISCO"
2140  ?:?:?:?
2150  ? "                              *******    A
TTENDI  ******"
2160  open "O",#1,n$
2170  ? #1,n
2180  for i%=1 to n
2190  ? #1,a$(i%)
2200  ? #1,d$(i%)
2210  ? #1,c(i%)
2220  next
2230  close
2240  ?:?:?:?
```

```
  2250  ? "                               CARICAMEN
TO EFFETTUATO"

  2260  ?:?

  2270  input "                              Premi <
RETURN> ... ";i$

  2280  goto 200

  2290  rem

  2300  rem ****************************

  2310  rem *                            *

  2320  rem *       FINE PROGRAMMA       *

  2330  rem *                            *

  2340  rem ****************************

  2350  rem

  2360  clearw 2

  2370  gotoxy 0,0

  2380  ?:?:?:?:?:?:?:?

  2390  ? "                        Sei sicuro di
 voler finire <s/n> ";


  2400  input i$

  2410  if i$="s" or i$="S" then clearw 2: end

  2420  goto 200

  2430  rem

  2440  rem ****************************

  2450  rem *                            *
```

```
2460  rem *  ROUTINE DI ERRORE DISCO   *
2470  rem *                            *
2480  rem ****************************
2490  rem
2500  clearw 2
2510  gotoxy 0,0
2520  close #1
2530  ?:?:?:?:?
2540  ? "                              FILE NON E
SISTENTE !!!!"
2550  ?:?:?:?:?
2560  ? "                              Premi <RET
URN> ... ";
2570  input i$
2580  resume 200
2590  goto 200
```

Programma n. 7

TITOLO : DISEGNI TRIDIMENSIONALI ANIMATI
LUNGHEZZA : 6604 Byte

DESCRIZIONE:

Il programma esegue la rappresentazione tridimensionale di una figura in tempo reale facendola ruotare intorno al suo baricentro. Costruendo una figura con un limitato numero di vertici e dando un angolo di rotazione (step) piccolo è possibile avere un effetto di animazione molto realistico. Dopo aver caricato il programma e averlo mandato in esecuzione, con RUN, verrà richiesto il numero di vertici di cui è composta la figura (da 1 a 100). Potrà sembrare strano il fatto di avere un solido composto ad esempio da due soli punti, ma tenendo conto che si può rappresentare come un piano ruotato nello spazio, i vantaggi sono tangibili. Viene richiesta anche l'altezza del solido (da 0 a 120) e quindi in sequenza le coordinate x e y (da 0 a 150). Alla fine di questa operazione bisogna introdurre l'angolo di rotazione (tra 0 e 360 gradi) che dovrà compiere la figura e il numero totale di rotazioni (da 1 a 100). Durante le rotazioni (la cui velocità dipende dal numero totale di vertici da gestire) il solido viene sempre inserito al centro dello schermo da una apposita routine che calcola in ogni momento le coordinate minime e massime di tutti i vertici. Una volta terminata la fase di animazione , premendo <RETURN>, si passa ad una fase di menù ad un livello inferiore nell'albero. Le opzioni disponibili sono cinque, vediamole nel dettaglio.

1. CORREZIONE COORDINATE

   Permette la correzione delle coordinate della figura. Viene chiesto il numero progressivo del vertice da correggere, quindi se si intende porre mano alle x o alle y. Viene proposto il valore attuale e richiesto quello corretto.

2. CORREZIONE ALTEZZA

   Come per l'opzione precedente solo che in questo caso si viene a correggere il valore dell'altezza del solido.

3. RIVEDERE IL DISEGNO

   Permette di cambiare l'angolo di rotazione della figura e il numero totale di rotazione da eseguirsi.

4. NUOVA FIGURA

   Permette l'inserimento delle coordinate di un nuovo solido.

5. FINIRE IL PROGRAMMA

   Conclude la fase di lavoro e ritorna al BASIC.

COMMENTO ALLE SINGOLE ISTRUZIONI	
10/160	Inizializzazione del programma e delle sue variabili.
170/520	Input dei dati e controllo sugli stessi.
530/840	Rotazione.
850/1060	Disegno tridimensionale.
1070/1250	Calcolo minimo e massimo.
1260/1520	Menù.
1530/1840	Correzione coordinate.
1850/2020	Correzione dell'altezza.
2030/2130	Pulizia delle finestre e fine programma.

ELENCO VARIABILI	
pi	pigreco
a()	matrice ascisse
b()	matrice ordinate
n	numero di dati
h	altezza del solido
i	ciclo
i$	lettura tastiera
st	step di rotazione
nr	numero di rotazioni
t	ciclo
m%	ciclo angolo di rotazione
ag	angolo
xa	coordinata di disegno
ya	coordinata di disegno
xm	coordinata massima
ym	coordinata massima
xn	coordinata minima
yn	coordinata minima
xt	coordinata di rotazione
yt	coordinata di rotazione
xp	coordinata di disegno
i%	ciclo di disegno
u	valore scelta menù
cc	coordinata da correggere
f	coordinata modificata
t%	ciclo pulizia finestre

```
10    rem   *******************************
20    rem   *                             *
30    rem   *        PROGRAMMA n. 7       *
40    rem   *                             *
50    rem   *     ANIMAZIONI FIGURE       *
60    rem   *      TRIDIMENSIONALI        *
70    rem   *                             *
80    rem   *******************************
90    rem
100   clear
110   fullw 2
120   clearw 2
130   pi=3.1415927
140   gosub 1070
150   dim a(100)
160   dim b(100)
170   rem
180   rem   *******************************
190   rem   *                             *
200   rem   *         INPUT DATI          *
210   rem   *                             *
220   rem   *******************************
230   rem
```

```
240   gotoxy 0,0

250   ?:?

260   ? "                    DISEGNI TRIDIMEN
SIONALI ANIMATI"
270   ? "                    ----------------
-------------------"

280   ?:?:?

290   ? "            Quanti dati vuoi caricar
e (1/100) ";

300   input "";n

310   n=int(abs(n))

320   if n>100 or n<1 then 290

330   ?:?

340   ? "            Dammi l:altezza (0/120)
";

350   input h

360   h=int(abs(h))

370   if h<0 or h>120 then 330

380   ?:?

390   for i=1 to n

400   ? "            Coordinata x(";i;")   (0/
150) ";

410   input "";a(i)

420   a(i)=int(abs(a(i)))

430   if a(i)>150 or a(i)<0 then 400

440   ? "            Coordinata y(";i;")   (0/
150) ";
```

```
450   input "";b(i)

460   b(i)=int(abs(b(i)))

470   if b(i)>150 or b(i)<0 then 440

480   ? "                       -----------------------------
--------"

490   next

500   ?:?

510   ? "                       Fine caricamento premi <
RETURN> ";

520   input "";i$

530   rem

540   rem   ******************************

550   rem   *                            *
560   rem   *          ROTAZIONE         *

570   rem   *                            *

580   rem   ******************************

590   rem

600   gosub 1070

610   clearw 2

620   gotoxy 0,0

630   ?:?:?:?

640   ? "                           ANIMAZIONE"

650   ?:?:?

660   ? "                      Dammi angolo di rotazione
(0/360) ";
```

```
670    input "";st

680    if st<0 or st>360 then 660

690    ?:?

700    input "                 Quante rotazioni (1/10
0) ";nr

710    if nr<1 or nr>100 then 700

720    for t=1 to nr

730    for m%=0 to 360 step st

740    ag=st*2*pi/360

750    gosub 1070

760    xa=(xm+xn)/2

770    ya=(ym+yn)/2

780    for i%=1 to n

790    yt=ya+(b(i%)-ya)*cos(ag)-(a(i%)-xa)*sin(a
g)

800    xt=xa+(b(i%)-ya)*sin(ag)+(a(i%)-xa)*cos(a
g)

810    b(i%)=yt

820    a(i%)=xt

830    next

840    gosub 1070

850    rem

860    rem    ******************************

870    rem    *                            *
```

```
880   rem   *   DISEGNO TRIDIMENSIONALE   *

890   rem   *                             *

900   rem   ******************************

910   rem

920   xp=(614-xm+xn)/2

930   clearw 2

940   for i%=1 to n-1

950   linef a(i%)+xp,300-b(i%),a(i%+1)+xp,300-b
(i%+1)

960   linef a(i%)+xp,300-h-b(i%),a(i%+1)+xp,300
-h-b(i%+1)

970   linef a(i%)+xp,300-b(i%),a(i%)+xp,300-h-b
(i%)

980   next

990   linef a(1)+xp,300-b(1),a(n)+xp,300-b(n)

1000  linef a(1)+xp,300-h-b(1),a(n)+xp,300-h-b(
n)

1010  linef a(n)+xp,300-b(n),a(n)+xp,300-h-b(n)

1020  next

1030  next

1040  ? "                 Premi <RETURN> per c
ontinuare ..... ";

1050  input "";i$

1060  goto 1260

1070  rem
```

```
1080  rem   *******************************
1090  rem   *                             *
1100  rem   *  CALCOLO MINIMO E MASSIMO   *
1110  rem   *                             *
1120  rem   *******************************
1130  rem
1140  xn=23e8
1150  xm=23e-8
1160  yn=23e8
1170  ym=23e-8
1180  for i%=1 to n
1190  if a(i%)>xm then xm=a(i%)
1200  if a(i%)<xn then xn=a(i%)
1210  if b(i%)>ym then ym=b(i%)
1220  if b(i%)<yn then yn=b(i%)
1230  next
1240  return
1250  goto 1250
1260  rem
1270  rem   *******************************
1280  rem   *                             *
1290  rem   *            MENU:            *
1300  rem   *                             *
```

```
1310  rem   ******************************
1320  rem
1330  clearw 2
1340  gotoxy 0,0
1350  ?:?
1360  ? "                             ELENCO OPZIONI
 DISPONIBILI"
1370  ?:?:?
1380  ? "                             1. Correzione
coordinate"
1390  ?
1400  ? "                             2. Correzione
altezza"
1410  ?
1420  ? "                             3. Rivedere di
segno"
1430  ?
1440  ? "                             4. Nuova figur
a"
1450  ?
1460  ? "                             5. Fine progra
mma "
1470  ?:?:?
1480  ? "                             Scegli (1/5) .
... ";
1490  input "";i$
```

```
  1500  if i$<"1" or i$>"5" then 1490

  1510  u=val(i$)

  1520  on u goto 1530,1850,530,90,2030

  1530  rem

  1540  rem   *******************************

  1550  rem   *                             *

  1560  rem   *    CORREZIONE COORDINATE    *

  1570  rem   *                             *

  1580  rem   *******************************

  1590  rem

  1600  clearw 2

  1610  gotoxy 0,0

  1620  ?:?:?:?

  1630  ?"                          CORREZIONE COORDI
NATE"

  1640  ?:?:?

  1650  ?"                          Quale vertice vuo
i correggere ( 1 / ";n;") ";

  1660  input "";cc

  1670  if cc<1 or cc>n then 1650

  1680  ?"                          Vuoi correggere <
x> o <y> ";

  1690  input i$

  1700  if i$<>"x" and i$<>"y" then 1680
```

```
1710 ?

1720 if i$="y" then 1790

1730 ? "                              Valore attuale di
x(";cc;") = ";a(cc)

1740 ? "                              Valore modificato
(0/150) = ";

1750 input f

1760 if f<0 or f>150 then 1740
1770 a(cc)=f

1780 goto 1260

1790 ? "                              Valore attuale di
y(";cc;") = ";b(cc)

1800 ? "                              Valore modificato
(0/150) = ";

1810 input f

1820 if f<0 or f>150 then 1800

1830 b(cc)=f

1840 goto 1260

1850 rem

1860 rem  ******************************

1870 rem  *                            *

1880 rem  *     CORREZIONE ALTEZZA     *

1890 rem  *                            *

1900 rem  ******************************

1910 rem
```

```
   1920  clearw 2

   1930  gotoxy 0,0

   1940  ?:?:?:?

   1950  ? "                  CORREZIONE ALTEZZ
A"

   1960  ?:?:?

   1970  ? "                  Valore attuale al
tezza    = ";h

   1980  ? "                  Valore modificato
 (0/120) = ";

   1990  input f

   2000  if f<0 or f>120 then 1980

   2010  h=f

   2020  goto 1260

   2030  rem

   2040  rem   ******************************

   2050  rem   *                            *

   2060  rem   *      FINE PROGRAMMA        *
   2070  rem   *                            *

   2080  rem   ******************************

   2090  rem

   2100  for t%=0 to 3

   2110  clearw t%

   2120  next

   2130  end
```

Programma n. 8

TITOLO : AGENDA PERSONALE APPUNTAMENTI
LUNGHEZZA : 6851 Byte

DESCRIZIONE:

Molto spesso è indispensabile avere a disposizione una agenda in grado di gestire con facilità e sicurezza i propri appuntamenti. Tanto meglio se elettronica e utilizzante il nostro ATARI. Quella presentata permette di archiviare fino a dieci appuntamenti per ogni giorno per un anno (3650 annotazioni !). Veniamo ad una descrizione di cosa accade dopo aver fatto girare il programma. Le possibilità offerte sono tre:

1. CREA/MODIFICA UN APPUNTAMENTO
   Permette la creazione o la modificazione di un appuntamento precedentemente inserito. È necessario introdurre il mese che viene immediatamente caricato da disco se presente.
   Viene chiesto quindi il giorno e segnalato se esistente oppure se si tratta di un nuovo inserimento. Esiste il controllo sulla correttezza del numero del giorno in relazione al mese (es. non è possibile chiedere il 30 Febbraio!). Si inserisce di seguito il numero di appuntamenti che si intende modificare e il loro numero progressivo. Si possono archiviare fino a 10 appuntamenti per ogni giorno. Se le modifiche sono terminate il programma procede ad un nuovo salvataggio dei dati.

2. RICERCA APPUNTAMENTI
   Analogamente all'opzione precedente viene richiesto il mese da visualizzare e successivamente il giorno. Se non esiste alcun appuntamento nel giorno selezionato, una scritta ci avverte e richiede un altro giorno.

3. ELENCO TUTTI APPUNTAMENTI
   Elenca in sequenza succinta gli appuntamenti di tutto un mese.

NOTA IMPORTANTE: Utilizzando il programma con alcune versioni del BASIC è possibile notare delle scorrettezze introdotte dal sistema operativo. È bene quindi , prima di utilizzare l'agenda, caricare anche il PROGRAMMA n. 9 di questo libro che crea un file iniziale eliminando queste incertezze. Si tratta di un file contenente tutti i mesi vuoti ma neccessari per una corretta lettura da parte del computer. Ricordo che per uscire dal programma è sufficiente premere contemporaneamente CONTROL + G.

COMMENTO SINGOLE ISTRUZIONI	
10/190	Inizializzazione variabili e caricamento delle matrici.
200/410	Menù principale.
420/1050	Crea nuovo appuntamento.
1060/1460	Ricerca un appuntamento.
1470/1870	Ricerca tutti gli appuntamenti.
1880/2020	Routine errore n. 1.
2030/2140	Routine errore n. 2.
2150/2320	Routine errore n. 3.

ELENCO VARIABILI	
a$()	matrice appuntamenti
m$()	matrice nome mesi
l()	matrice lunghezza mesi
i%	ciclo
i$	scelta da menù
i	valore lettura tastiera
t%	ciclo
y%	ciclo
j%	ciclo
m	mese
g	giorno
nm	numero appuntamenti
ap	numero appuntamento
j	ciclo
e%	ciclo

```
10    rem ********************************
20    rem *                              *
30    rem *        PROGRAMMA n. 8        *
40    rem *                              *
50    rem * AGENDA PERSONALE APPUNTAMENTI *
60    rem *                              *
70    rem ********************************
```

```
80    rem
90    fullw 2
100   clearw 2
110   gotoxy 0,3
120   dim a$(10,31)
130   dim m$(12)
140   dim l(12)
150   restore
160   for i%=1 to 12
170   read m$(i%)
180   read l(i%)
190   next
200   rem
210   rem ****************************
220   rem *                          *
230   rem *     MENU: PRINCIPALE     *
240   rem *                          *
250   rem ****************************
260   rem
270   clearw 2
280   gotoxy 0,3
290   ? "                                    AGENDA PER
SONALE"
300   ?:?:?
```

```
310    ? "                              1. Crea/Modific
a un appuntamento"

320    ?

330    ? "                              2. Ricerca appu
ntamenti"

340    ?

350    ? "                              3. Elenco tutti
 appuntamenti"

360    ?:?:?

370    ? "                              Dammi il numero
 <1/3>    ";

380    input "";i$

390    if i$<"1" or i$>"3" then 380

400    i=val(i$)

410    on i goto 420,1060,1470

420    rem

430    rem *****************************

440    rem *                           *

450    rem *  CREA NUOVO APPUNTAMENTO  *

460    rem *                           *

470    rem *****************************

480    rem

490    clearw 2

500    gotoxy 0,3

510    erase a$
```

```
520   dim a$(10,31)

530   for t%=1 to 31

540   for y%=1 to 10

550   a$(y%,t%)="  "

560   next

570   next

580   ? "                      CREA UN APPUNTAMEN
TO"

590   ?:?

600   ? "                      Dammi il mese ( 1
/ 12 ) ";

610   input m

620   if m<1 or m>12 then 600

630   on error goto 2030

640   open "I",#1,m$(m)

650   for i%=1 to 31

660   for j%=1 to 10

670   input #1,a$(j%,i%)

680   next

690   next

700   close #1

710   ?

720   ? "                      Dammi il giorno (
1 /";l(m);") ";
```

```
730   input g
740   if g<1 or g>l(m) then 760
750   goto 800
760   ?
770   ? "                    NON esiste il";g;"
";m$(m);" !!!"
780   ?:?
790   goto 710
800   for i=1 to 10
810   ? using "###";i;
820   ? " > ";a$(i,g)
830   next
840   ? "          ----------------------------------
----------------------"
850   ?
860   ? "          Quanti appuntamenti vuoi mod
ificare  (1/10)  ";
870   input "";nm
880   if nm>10 or nm<1 then 860
890   for i=1 to nm
900   input "          Appuntamento da modifica
re (1/10) ";ap
910   if ap<1 or ap> 10 then 900
920   input "   >  ";a$(ap,g)
930   next
```

```
940   input "           Vuoi modificare ancora (
s/n)  ";i$

950   if i$<>"n" and i$<>"s" then 940

960   if i$="s" then 710

970   ?:? "          ----- Salvataggio in corso
 -----"

980   open "O",#1,m$(m)

990   for i=1 to 31

1000  for j=1 to 10

1010  ? #1,a$(j,i)

1020  next

1030  next

1040  close #1

1050  goto 150

1060  rem
1070  rem ****************************

1080  rem *                          *

1090  rem *  RICERCA UN APPUNTAMENTO  *

1100  rem *                          *

1110  rem ****************************

1120  rem

1130  clearw 2

1140  gotoxy 0,3

1150  ? "                        RICERCA APPU
NTAMENTI"
```

```
1160  ?:?:?

1170  erase a$

1180  dim a$(10,31)

1190  ? "                              Quale mese vuoi v
edere (1/12) ";

1200  input "";m

1210  if m<1 or m>12 then 1190

1220  on error goto 1880

1230  open "I",#1,m$(m)

1240  for i%=1 to 31

1250  for j%=1 to 10

1260  input #1,a$(j%,i%)

1270  next

1280  next

1290  close #1

1300  input "                              Dammi il gior
no ";g

1310  if g<1 or g>l(m) then 1300

1320  for i%=1 to 10

1330  if len(a$(i%,g))>2 then 1370

1340  next

1350  ? "                --- nessun appuntamento
 il";g;m$(m);" ---"

1360  goto 1300
```

```
1370  ? "                              Appuntamenti del
";g;m$(m)

1380  ?

1390  for i%=1 to 10

1400  if len(a$(i%,g))>2 then ? "    >  ";a$(i%,
g)

1410  next

1420  ?

1430  ? "                              Vuoi cercare anco
ra <s/n> ... ";

1440  input i$

1450  if i$<>"s" and i$<>"n" then 1440

1460  if i$="s" then 1300 else 150

1470  rem

1480  rem ********************************

1490  rem *                              *

1500  rem *  RICERCA TUTTI APPUNTAMENTI   *

1510  rem *                              *

1520  rem ********************************

1530  rem

1540  clearw 2

1550  gotoxy 0,3

1560  ? "                              RICERCA TUTTI GLI
 APPUNTAMENTI"
```

```
1570  ?:?:?

1580  ?  "                           Quale mese vuoi v
edere (1/12) ";

1590  input "";m

1600  if m<1 or m>12 then 1190

1610  erase a$

1620  dim a$(10,31)

1630  on error goto 2150

1640  open "I",#1,m$(m)

1650  for i%=1 to 31

1660  for j%=1 to 10

1670  input #1,a$(j%,i%)

1680  next

1690  next

1700  close #1

1710  for g=1 to l(m)

1720  for e%=1 to 10

1730  if len(a$(e%,g))>2 then 1760

1740  next e%

1750  goto 1820

1760  ?

1770  ?  "    ----------- Giorno : ";g;" ";m$(m)
;"-----------"

1780  ?
```

```
1790  for i%=1 to 10
1800  if len(a$(i%,g))>2 then ? "    ";a$(i%,g)
1810  next i%
1820  next g
1830  ?
1840  ? "                         Vuoi cercare anco
ra <s/n> ... ";
1850  input i$
1860  if i$<>"s" and i$<>"n" then 1840
1870  if i$="s" then 1470 else 150
1880  rem
1890  rem *****************************
1900  rem *                           *
1910  rem *    ROUTINE ERRORE N. 1    *
1920  rem *                           *
1930  rem *****************************
1940  rem
1950  close #1
1960  if err=62 then resume 2200: goto 1300
1970  ? "                         NON esiste il fil
e <";m$(m);">  !!"
1980  ?:?
1990  ? "                         Premi <RETURN> pe
r continuare ... ";
```

```
2000 input "";i$
2010 resume 150
2020 goto 150
2030 rem
2040 rem *****************************
2050 rem *                           *
2060 rem *    ROUTINE ERRORE N. 2    *
2070 rem *                           *
2080 rem *****************************
2090 rem
2100 close #1
2110 if err=62 then resume 1500: goto 710
2120 ? "                         ---  nessun file
<";m$(m);">  ---"
2130 resume 710
2140 goto 710
2150 rem
2160 rem *****************************
2170 rem *                           *
2180 rem *    ROUTINE ERRORE N. 3    *
2190 rem *                           *
2200 rem *****************************
2210 rem
2220 close #1
```

```
  2230  if err=62 then resume 3200: goto 1710

  2240  ? "                        NON esiste il fil
e <";m$(m);">  !!"

  2250  ?:?

  2260  ? "                        Premi <RETURN> pe
r continuare ... ";

  2270  input "";i$

  2280  resume 150

  2290  goto 150

  2300  data GENNAIO,31,FEBBRAIO,29,MARZO,31,APRI
LE,30

  2310  data MAGGIO,31,GIUGNO,30,LUGLIO,31,AGOSTO
,31

  2320  data SETTEMBRE,30,OTTOBRE,31,NOVEMBRE,30,
DICEMBRE,31
```

Programma n.9

TITOLO : GENERATORE FILE PER AGENDA
LUNGHEZZA : 940 Byte

DESCRIZIONE:

Il programma, come accennato nel programma precedente, genera un file di mesi per il corretto funzionamento dell'agenda. Bisogna introdurre nel drive un dischetto formattato e non protetto. Si lancia quindi il programma che visualizza sullo schermo lo stato di avanzamento della creazione. Il programma cancella tutti gli altri file con lo stesso nome che sono contenuti nel disco.

COMMENTO ALLE SINGOLE ISTRUZIONI	
10/130	Inizializzazione programma e pulizia delle finestre.
140/260	Caricamento dei file.
270/290	Dati da caricare.

ELENCO VARIABILI	
t	ciclo lettura mesi
m$	nome del mese
l	lunghezza del mese
j	ciclo caricamento appuntamenti
i	ciclo caricamento giorni

```
10   rem ********************************
20   rem *                              *
30   rem *        PROGRAMMA n. 9        *
40   rem *                              *
50   rem * GENERATORE FILE PER AGENDA   *
60   rem *                              *
70   rem ********************************
```

```
80    rem

90    fullw 2

100   clearw 2

110   gotoxy 0,3

120   ? "                                   ----------------
ATTENDI -------------"

130   ?:?

140   for t=1 to 12

150   read m$,l

160   ? "                                   Generazione mese
 :  ";m$

170   open "O",#1,m$

180   for j=1 to 10

190   for i=1 to l

200   ? #1," > "

210   next

220   next

230   close #1

240   next

250   ?:?

260   ? "                                   ---------- HO TE
RMINATO -------------"
```

```
   270    data GENNAIO,31,FEBBRAIO,29,MARZO,31,APRI
LE,30

   280    data MAGGIO,31,GIUGNO,30,LUGLIO,31,AGOSTO
,31

   290    data SETTEMBRE,30,OTTOBRE,31,NOVEMBRE,30,
DICEMBRE,31
```

Programma n.10

TITOLO : CALCOLO BARICENTRO MASSE
LUNGHEZZA : 6054 Byte

DESCRIZIONE:

Il programma esegue il calcolo e la rappresentazione grafica di una serie di aree (massimo 20) e dei relativi centri di massa.

Si tratta di un programma che trova ampie applicazioni nel campo della statica, della scienza delle costruzioni eccetera. Le aree sono rettangolari ma combinandole opportunamente è possibile ottenere qualsiasi figura anche molto complessa. Il programma utilizza la massima risoluzione possibile indicando anche le coordinate del centro di massa totale. Il simbolo del centro (G) è ottenuto in alta risoluzione collegando fra loro opportuni segmenti. Tutte le aree risultano riempite con l'istruzione FILL ed è indicato di ognuna il baricentro. Il programma è strutturato ad albero su due menù. Vediamo in dettaglio cosa accade dopo aver mandato in esecuzione il programma.

MENÙ PRINCIPALE

1. CARICAMENTO DATI

   Consente il caricamento delle coordinate di una nuova serie di aree. Viene richiesto il numero delle figure da introdurre (1/20) e di ognuna di esse le ascisse (0/400), le ordinate (0/250), la base (0/250) ed infine l'altezza (0/250). Per la posizione delle coordinate e delle dimensioni relative (base e altezza) si rimanda il lettore alla figura seguente.

2. DEMO

   Carica nella memoria una serie di dati precedentemente inseriti a scopo dimostrativo e per acquisire maggiore dimestichezza nell'uso del programma.

   Sia la prima che la seconda opzione portano ad un menù successivo nel quale sono presenti 4 possibili scelte.

MENÙ SECONDARIO

1. DISEGNO AREE E BARICENTRO

   Esegue il disegno in alta risoluzione delle aree precedentemente inserite, riporta graficamente la posizione del centro di massa e le coordinate in forma numerica. Per continuare premere <RETURN>.

2. CALCOLO BARICENTRI AREE

   Riporta in forma tabellare il numero progressivo dell'area, le dimensioni di base e altezza, la superficie e le coordinate baricentriche di ogni area e quelli totali.

3. CARICAMENTO ALTRE AREE

   Riporta il programma al menù principale.

4. FINIRE PROGRAMMA

   Esce dal programma dopo una conferma da parte dell'operatore.

COMMENTO ALLE SINGOLE ISTRUZIONI	
10/190	Inizializzazione programma e matrici, presentazione.
200/280	Menù principale.
290/500	Caricamento coordinate aree.
510/640	Menù secondario.
650/1120	Disegno aree e baricentro.
1130/1480	Calcolo e stampa baricentri aree.
1490/1610	Fine programma.
1620/1800	Calcolo baricentri.
1810/1970	Demo.

ELENCO VARIABILI			
x()	matrice ascisse	yi	ordinata di lavoro
y()	matrice ordinate	q	variabile di posizione
b()	matrice basi	r	coefficiente moltiplicatore
h()	matrice altezze	u	coefficiente moltiplicatore
i$	lettura tastiera	xa	coordinata di disegno
k	valore lettura tastiera	ya	coordinata di disegno
n	numero di aree	i	ciclo sarea totale
a$	lettura tastiera	sx	sommatoria x
xg	ascissa baricentro	sy	sommatoria y
yg	ordinata baricentro	p	ciclo demo
xi	ascissa di lavoro		

```
10      rem ******************************
20      rem *                            *
30      rem *      PROGRAMMA n. 10       *
40      rem *                            *
50      rem *  CALCOLO BARICENTRO MASSE  *
60      rem *                            *
70      rem ******************************
80      rem
90      fullw 2
100     clearw 2
110     gotoxy 0,3
120     dim x(20)
130     dim y(20)
140     dim b(20)
150     dim h(20)
160     ? "                              CALCOLO BA
RICENTRO MASSE"
170     gotoxy 0,12
180     ? "                         Premi <RETURN>
 per continuare ...";
190     input i$
200     clearw 2
210     gotoxy 0,5
220     ? "                         1. Caricamento
 dati"
```

```
230   ?:? "                         2. Demo"

240   ?:?:?"                        Scegli <1/2
> .. ";

250   input i$

260   if i$<"1" or i$>"2" then 200

270   k=val (i$)

280   on k goto 290,1800

290   clearw 2

300   gotoxy 0,3

310   ? "                         Caricamento co
ordinate aree"

320   ?:?:?

330   input "                        Quante are
e <1/20> ";n

340   if n>10 or n<1 then 330

350   ?:?

360   for i=1 to n

370   input "                        Dammi la x
 (0/400) ";x(i)

380   if x(i)>400 or x(i)<0 then 370

390   input "                        Dammi la y
 (0/250) ";y(i)

400   if y(i)>250 or y(i)<0 then 390

410   input "                        Dammi la b
 (0/400) ";b(i)

420   if x(i)+b(i)>400 then 410
```

```
  430   input "                    Dammi la h
 (0/250) ";h(i)

  440   if y(i)-h(i)<0 then 390

  450   ? "                  ---------------
----------------------------------"

  460   ?

  470   next

  480   ?:?

  490   ? "                  Premi <RETURN>
 per continuare ... ";


  500   input i$

  510   clearw 2

  520   gotoxy 0,3

  530   ? "                  ------- OPZION
I DISPONIBILI ------- "


  540   ?:?

  550   ? "                  1. Disegno are
e e baricentro"

  560   ?:? "                   2. Calcolo b
aricentri aree"

  570   ?:? "                   3. Caricamen
to altre aree"

  580   ?:? "                   4. Fine prog
ramma"

  590   ?:?
```

```
   600   ? "                              Scegli <1/4> .
.. ";
   610   input a$
   620   if a$<"1" or a$>"4" then 600
   630   k=val(a$)
   640   on k goto 650,1130,200,1490
   650   rem
   660   rem *******************************
   670   rem *                            *
   680   rem *  DISEGNO AREE E BARICENTRO  *
   690   rem *                            *
   700   rem *******************************
   710   rem
   720   color 1,1,1,1,2
   730   clearw 2
   740   gotoxy 0,0
   750   gosub 1620
   760   ? "                              Aree e bari
centri"
   770   gotoxy 0,19
   780   ? "          Coordinate    XG=";
   790   ? using "###";xg;
   800   ? "         YG=";
   810   ? using "###";yg;
```

```
820    xi=100

830    yi=47

840    q=344

850    linef xi,yi,xi+400,yi

860    linef xi,yi+250,xi+400,yi+250

870    linef xi+400,yi,xi+400,yi+250

880    linef xi,yi,xi,yi+250

890    for i=1 to n
900    linef xi+x(i),q-yi-y(i),xi+x(i)+b(i),q-yi
-y(i)

910    linef xi+x(i),q-yi-y(i)+h(i),xi+x(i)+b(i)
,q-yi-y(i)+h(i)

920    linef xi+x(i),q-yi-y(i),xi+x(i),q-yi-y(i)
+h(i)

930    linef xi+x(i)+b(i),q-yi-y(i),xi+x(i)+b(i)
,q-yi-y(i)+h(i)

940    if b(i)>2 and h(i)>2 then fill xi+x(i)+2,
q-yi-y(i)+2

950    linef xi+x(i)+b(i),q-yi-y(i),xi+x(i),q-yi
-y(i)+h(i)

960    linef xi+x(i),q-yi-y(i),xi+x(i)+b(i),q-yi
-y(i)+h(i)

970    next

980    r=8

990    u=5

1000   linef xi+xg-r,q-yi-yg,xi+xg+r,q-yi-yg

1010   linef xi+xg,q-yi-yg-r,xi+xg,q-yi-yg+r
```

```
1020  xa=xi+xg+25
1030  ya=q-yi-yg-25
1040  linef xa,ya,xa-u,ya
1050  linef xa-u,ya,xa-2*u,ya+u
1060  linef xa-2*u,ya+u,xa-2*u,ya+2*u
1070  linef xa-2*u,ya+2*u,xa-u,ya+3*u
1080  linef xa-u,ya+3*u,xa,ya+3*u
1090  linef xa,ya+3*u,xa,ya+1.5*u
1100  linef xa,ya+1.5*u,xa-u,ya+1.5*u
1110  input "     >> ";i$
1120  goto 510
1130  rem
1140  rem ****************************
1150  rem *                          *
1160  rem *  CALCOLO BARICENTRI AREE  *
1170  rem *                          *
1180  rem ****************************
1190  rem
1200  clearw 2
1210  gotoxy 0,0
1220  ? "      Area     base     altezza    area
 xg        yg"
1230  ? "       ------------------------------------------------
-----------------";
1240  ?
```

```
1250 for i=1 to n
1260 ? "     ";
1270 ? using "##";i;
1280 ? "      ";
1290 ? using "###";b(i);
1300 ? "     ";
1310 ? using "###";h(i);
1320 ? "     ";
1330 ? using "######";b(i)*h(i);
1340 ? "    ";
1350 ? using "###";x(i)+b(i)/2;
1360 ? "    ";
1370 ? using "###";y(i)-h(i)/2
1380 next
1390 gosub 1620
1400 ? "     Area totale =           ";
1410 ? using "######";s;
1420 ? "    ";
1430 ? using "###";xg;
1440 ? "    ";
1450 ? using "###";yg
1460 ?:? "                    Premi <RETURN> per
continuare ..  ";
```

```
1470  input i$

1480  goto 510

1490  rem

1500  rem ****************************

1510  rem *                          *

1520  rem *     FINE PROGRAMMA       *

1530  rem *                          *

1540  rem ****************************

1550  rem

1560  clearw 2

1570  gotoxy 0,4

1580  ? "                       Sei sicuro di
voler finire <s/n> ";

1590  input i$

1600  if i$="s" then clearw 2: end

1610  goto 510

1620  rem

1630  rem ****************************

1640  rem *                          *

1650  rem *    CALCOLO BARICENTRO    *

1660  rem *                          *

1670  rem ****************************

1680  rem
```

```
1690  s=0
1700  sx=0
1710  sy=0
1720  for j=1 to n
1730  s=s+b(j)*h(j)
1740  sx=sx+b(j)*h(j)*(x(j)+b(j)/2)
1750  sy=sy+b(j)*h(j)*(y(j)-h(j)/2)
1760  next
1770  xg=sx/s
1780  yg=sy/s
1790  return
1800  rem
1810  rem ***************************
1820  rem *                         *
1830  rem *          DEMO           *
1840  rem *                         *
1850  rem ***************************
1860  rem
1870  restore
1880  n=5
1890  for p=1 to n
1900  read x(p),y(p),b(p),h(p)
```

Elenco tasti funzione

Quando è inserito il BASIC i tasti funzione assumono il seguente significato:

F1 : INSERISCI SPAZIO
F2 : CANCELLA CARATTERE
F3 : INSERISCI LINEA
F4 : CANCELLA LINEA
F5 : PAGINA SU
F6 : PAGINA GIÙ
F7 : CARICA TESTO DA DISCO
F8 : SALVA TESTO SU DISCO
F9 : NEW BUFFER
F10 : USCITA DALL'EDIT

APPENDICE

SPECIFICHE TECNICHE COMPUTER

Dei due ATARI, il 520 ST è dotato di un processore tipo MC68000 con 32 bit interni e 16 per i collegamenti con l'esterno. Viene controllato con una frequenza di clock pari a 8 Mhz. La memoria RAM è di 512 K (pari a 524288 Byte) e 16 KRom (16384 Byte).
La risoluzione grafica è selezionabile tra 640 x 400 in monocromatico, 320 x 200 con 16 colori, 640 x 200 con 4 colori.
La scelta può avvenire su una tavolozza di 512. La capacità dei dischetti di 3 pollici e mezzo è di 360 KByte formattati.
Il computer dispone della possibilità di collegarsi al MIDI, ad un monitor (RBG), ad una porta parallela (Centronics), seriale (RS232C), ai floppy disk, porta per Hard Disk (con velocità di trasferimento di 10 Mbyte al secondo), porta per cartuccia ROM (con capacità massima di 128 KByte), porta per joystick e mouse. Possiede un generatore sonoro a 3 voci con capacità di frequenze da 30 Hz alla soglia dell'udibile. La tastiera possiede 94 tasti ed è controllata da un processore 6301. Deve essere alimentato con +5V 2A, +12V 30mA, -12V 30mA.

APPENDICE

SPECIFICHE TECNICHE DISK DRIVE SF354

Il disk drive consente di immagazzinare fino a 360 KByte formattati di dati con una densità di 135 tracce al pollice.
Ogni disco possiede 80 tracce. Il posizionamento della testina di lettura avviene con motori passo passo. La velocità di trasferimento delle informazioni è di 250 Kbit al secondo. La temperatura di funzionamento è compresa ta i 5 e i 45 gradi centigradi e opera con un'umidità relativa compresa tra il 20 e l'80 % . I dischi devono essere alimentati con 12V 60mA.

I computer Atari 520ST e 1040ST si presentano come macchine destinate a lasciare il segno nel mondo dei personal computer La facilità estrema di utilizzo unita a caratteristiche eccellenti e un basso costo costituiscono il piatto forte di queste macchine destinate al successo in qualsiasi ambiente, sia professionale che didattico. Questo libro insegna ad usarle anche non disponendo di alcuna conoscenza nel campo informatico. I programmatori più attenti possono trovare, invece, un valido aiuto nell'accurata descrizione dei comandi (sempre seguiti da numerosi esempi) e nei dieci programmi allegati e ampiamente documentati. I programmi affrontano i problemi più disparati rimanendo sempre nel campo professionale a cui è destinata la macchina. Durante la lettura il lettore viene seguito passo passo da opportuni esempi di applicazione pratica di quanto descritto.

ISBN 88-7056-686-2
9 788870 566864

L. 28.000

Cod. CC471

GRUPPO EDITORIALE JACKSON 471

Maurizio Galluzzo

il Manuale dell'ATARI 520 ST e 1040 ST